Communications of the Association for Information Systems

Volume 4 Article 15

December 2000

MOTC with Examples: An Interactive Aid for Multidimensional Hypothesis Generation

K. Balachandran
University of Pennsylvania, kb@grace.wharton.upenn.edu

J. Buzydlowski
University of Pennsylvania, jb@grace.wharton.upenn.edu

G. Dworman
University of Pennsylvania, gd@grace.wharton.upenn.edu

S.O. Kimbrough
University of Pennsylvania, sok@grace.wharton.upenn.edu

T. Shafer
University of Pennsylvania, ts@grace.wharton.upenn.edu

See next page for additional authors

MOTC with Examples: An Interactive Aid for Multidimensional Hypothesis Generation

Authors
K. Balachandran, J. Buzydlowski, G. Dworman, S.O. Kimbrough, T. Shafer, and W. Vachula

Communications of the Association for Information Systems

Volume 4, Article 15
December 2000

MOTC WITH EXAMPLES: AN INTERACTIVE AID FOR MULTIDIMENSIONAL HYPOTHESIS GENERATION

K. Balachandran, J. Buzydlowski, G. Dworman,
S.O. Kimbrough, T. Shafer, W. Vachula
The Wharton School
University of Pennsylvania

Kimbrough@Wharton.upenn.edu

DATA MINING

MOTC WITH EXAMPLES: AN INTERACTIVE AID FOR MULTIDIMENSIONAL HYPOTHESIS GENERATION

K. Balachandran, J. Buzydlowski, G. Dworman,
S.O. Kimbrough, T. Shafer, W. Vachula
The Wharton School
University of Pennsylvania

Kimbrough@Wharton.upenn.edu

ABSTRACT

This paper reports on conceptual development in the areas of database mining, and knowledge discovery in databases (KDD). Our efforts have also led to a prototype implementation, called MOTC, for exploring hypothesis space in large and complex data sets. Our KDD conceptual development rests on two main principles. First, we use the crosstab representation for working with qualitative data. This is by now standard practice in OLAP (on-line analytical processing) applications and we reaffirm it with additional reasons. Second, and innovatively, we use Prediction Analysis as a measure of goodness for hypotheses. Prediction Analysis is an established statistical technique for analysis of associations among qualitative variables. It generalizes and subsumes a large number of other such measures of association, depending upon specific assumptions the user is willing to make. As such, it provides a very useful framework for exploring hypothesis space in a KDD context. The paper illustrates these points with an extensive discussion of MOTC.

Keywords: knowledge data discovery, data mining, MOTC, decision support systems, prediction analysis, hypothesis generation

I. INTRODUCTION

Databases are underexploited. Typically created to record and facilitate business transactions, databases often contain valuable information which fails to be recognized and used by the organizations that own and maintain them. Such, at least, is a widespread belief. This has led to a burgeoning industry of research papers, start-up firms, and professional seminars, focusing on what has come to be called KDD (knowledge discovery in databases; see [Fayyad et al., 1996] for a collection of representative papers; and the annual KDD conference for the latest work: http://www.aaai.org/Conferences/KDD/kdd.html). Real money is being spent and much sweat is being produced in bets that valuable knowledge is there to be discovered and that software innovations will help discover and exploit this knowledge economically.

We share the widespread belief in the efficacy, or at least potential, of KDD, and are exploring a concept that-we believe-addresses a central problem in KDD, viz., hypothesis generation. In what follows we describe our concept and our implementation of it in a prototype system called MOTC. First, however, some comments to set the context.

The premise of KDD is that software innovations can materially contribute to more effective exploitation of databases. But just how can KDD software do this and what is its relation to standard statistical methods? Put bluntly, here is a question we have heard posed by many statisticians and statistically-trained practitioners: What does KDD have to offer that isn't done well already by multiple regression techniques? Put briefly, the answer is “plenty.” Standard

statistical methods, including regression analysis, are hypothesis *testing* methods. For example, what regression analysis does is accept a functional form for a model/hypothesis and then find the "best" instance of a model/hypothesis of that form. Even if we were to grant that computational approaches could never improve on this basic statistical task, very much remains to be done-and to be researched-in the interests of effective KDD.

Examples of "non-statistical" issues in KDD include the following.

1. Data cleaning

   What can be done to locate and ameliorate the pervasive problems of invalid or incomplete data?

2. "First cut" analysis

   What can be done to automatically provide an initial assessment of the patterns and potentially useful or interesting knowledge in a database? The aim here is, realistically, to automate *some* of the basic work that is now done by skilled human analysts.

3. Hypothesis generation

   What can be done to support, or even automate, the finding of plausible hypotheses in the data? Found hypotheses would, of course, need to be tested subsequently with statistical techniques, but where do you get "the contenders" in the first place?

Our attention, and the research results we are reporting in this paper, have focused on the hypothesis generation problem for KDD. Because hypothesis space is generally quite large (more on this below), it is normally quite impossible to enumerate and investigate all the potentially interesting hypotheses. Heuristics are necessary and, it would seem, a decision support philosophy is called for. What, then, are the main requirements, or desired features, of a decision support

tool for investigating hypothesis space? We identify the following as among the principal requirements. Such a tool should:

1. Support users in hypothesizing relationships and patterns among the variables in the data at hand (we call this *hypothesis hunting*).

2. Provide users with some indication of the validity, accuracy, and specificity of various hypotheses (*hypothesis evaluation*).

3. Provide effective visualizations for hypotheses, so that the powers of human visual processing can be exploited for exploring hypothesis space.

4. Support automated exploration of hypothesis space, with feedback and indicators for interactive (human-driven) exploration.

5. Support all of the above for data sets and hypotheses of reasonably high dimensionality, say between 4 and 200 dimensions, as well on large data sets (e.g., with millions of records).

What is needed, conceptually, to build such a tool?

1. A general concept or representation for data, hypotheses, and hypothesis space. This representation need not be universal, but should be broadly applicable. We call this the *hypothesis representation,* and we discuss it in Section 2.

2. Given a hypothesis representation, we also need an indicator of quality for the hypothesis in question. We call this the *measure of goodness,* and we discuss it in Section 3.

3. The hypothesis representation and the measure of goodness should fit with, cohere with, the requirements (and implicit goals, described above) of a DSS for exploring hypothesis space. We discuss our efforts and results in this regard in Sections 4-5.

II. HYPOTHESIS REPRESENTATION

There are three main elements to our hypothesis representation concept:

1. Focus on qualitative data.

2. Use the crosstab (also known as: data cube, multidimensional data, cross classifications of multivariate data) form for data (rather than, say, the relational form as in relational databases).

3. Represent hypotheses by identifying error values in the cells of the multidimensional (crosstab) data form.

These aspects of the concept, and why we have them, are perhaps best understood through a specific example.[1] Suppose we have data on two variables: X_1, party affiliation, and X_2, support for an increased government role in social services. X_1 can take on the following values: *Dem, Ind,* and *Rep* (Democrat, Independent, and Republican). X_2 can have any of the following values: *left, left-center, center, right-center, right.* Suppose we have 31 observations of the two variables taken together, as follows in Table 1.[2]

Focus on qualitative data. The variables X_1 and X_2 in Table 1 are qualitative (also known as: categorical) because they take on discrete values (three such values in the case of X_1 and five for X_2). X_1 is arguably a *nominal* variable because there is no compelling natural ordering for its three values.[3] *Dem* for example is neither more nor less than *Ind.* Similarly, in a business database *Sales-Region* and *Division* are nominal because, e.g., *Mid-Atlantic* is neither more

[1] The example that follows is from [Hindebrand et al., 1977]. We invite the reader to examine that discussion as a way of following up on this paper.

[2] We use the two-variable case for illustration only. As noted above, an important requirement for a hypothesis exploration DSS is that it handle reasonably high-dimensionality hypotheses. Except where noted, e.g., limitations of screen space in MOTC-like implementations-our points and methods generalize to arbitrarily many dimensions, at least in principle.

[3] Nothing much turns on this. One could argue that, at least for certain purposes, this is an ordinal variable. No matter. Our point is that this approach can handle nominal variables, if there are any.

Table 1: Party Affiliation and Support for Social Services by Top-Level Bureaucrats in Social Service Agencies ([Hildebrand et al., 1977a,p. 11])

Support	*Party Affiliation*			
	Dem	*Ind*	*Rep*	
Left	12	3	1	16
Left-center	1	2	2	5
Center	0	3	4	7
Right-center	0	1	1	2
Right	0	0	1	1
	13	9	9	31

nor less than *New England* and *Marketing* is neither more nor less than *Manufacturing.* X_2 on the other hand is an *ordinal* variable because there is a natural ordering for the values it takes on: *left, left-center, center* and so on. Similarly, in a business database, *Quarter* (*first, second, third, fourth*) is naturally ordered and therefore ordinal. If a variable, e.g., *Sales,* is quantitative, then (for our framework) it will have to be quantized, or *binned*. Thus, for example, *Sales* (V_2) might be binned as follows into five categories or bins (also known as: *forms* [Jambu, 1991]):[4]

$\mathbf{V_2^1}$
[0 - 20,000)
$\mathbf{V_2^2}$
[20,000 - 40,000)
$\mathbf{V_2^3}$
[40,000 - 60,000)

[4] How a basically quantitative variable should be binned-including how many forms it should have-is typically determined by the investigator, although some principles for automatic binning are available [Wand, 1997]. It is well known that infelicitous binning can lead to anomalies and distortions. In general for a quantitative variable it is better to have more bins than fewer, in order to reduce or even eliminate loss of information. Having more bins does have increased computational cost. Neglecting computational costs, Prediction Analysis transparently accommodates arbitrarily large numbers of bins (and cells); in particular it is unaffected by the presence of crosstab cells without data instances.

V_2^4 [60,000 - 80,000)

V_2^5 [80,000 +]

By way of justification for this assumed focus, we note the following: (1) Many variables, perhaps the majority, occurring in business databases are naturally qualitative; (2) A general framework, including both qualitative and quantitative variables, is highly desirable; (3) With felicitous binning quantitative variables can typically be represented qualitatively to a degree of accuracy sufficient for exploratory purposes; and (4) Transformation of inherently qualitative variables to a quantitative scale is inherently arbitrary and is known to induce results sensitive to the transformation imposed.

Use the crosstab form for data. This aspect of our focus requires less explanation and justification, since it is also standard practice in OLAP (on-line analytical processing) applications (cf., [Inmon, 1996,p. 179] on "the `cube' foundation for multi-dimension DBMS datamarts"; [Dhar and Stein, 1997,p. 45] on "hypercube data representations"; [Menninger, 1995] and [Codd et al., 1993] on "cubes"). Our reasons for using the crosstab form for data representation are simple and essentially identical to why it is now used so widely in OLAP applications (and has long been essential in statistics): the crosstab form easily accommodates qualitative variables and (most importantly) it has been demonstrated to be a natural representation for the sorts of reports and hypotheses users-managers and scientists-typically are interested in.[5] (See also the literature on information visualization. For a review see [Jones, 1995].)

Represent hypotheses by identifying error values in the cells of the multidimensional data form. Recalling our example data, in Table 1, suppose that an investigator has a hypothesis regarding how each bureaucrat's party affiliation

[5] We do not want to suggest that the data format evident in Table 1 is the only kind of crosstab representation for qualitative data. It isn't and the methods we discuss here, including MOTC itself, are not limited to this particular format, but from elaborating upon the point would be a

predicts the bureaucrat's support for increased social services. Following the notation of [Hildebrand et al., 1977a,Hildebrand et al., 1977b], we use the statement x ~> y to mean, roughly, "if x then predict y" or "x tends to be a sufficient condition for y."[6] Suppose our investigator's hypothesis, or prediction (call it P_1), is that Democrats tend to be left or left-center, Independents tend to be at the center, and Republicans tend to be center, right-center, or right. Equivalently, but more compactly, we can say:

> P_1: *Dem* ~> (*left* or *left-center*) and *Ind* ~> *center* and *Rep* ~> (*center* or *right-center* or *right*)

Equivalently, and in tabular form, we can label cells in the crosstab representation as either predicted by P_1, in which case they receive an error value of 0, or as not predicated by P_1, in which case they receive an error value of 1. Table 2 presents P_1 in this form.

Table 2: Error-cell representation for the hypothesis, or prediction, P_1.

Support	*Party Affiliation*		
	Dem	*Ind*	*Rep*
Left	0	1	1
Left-center	0	1	1
Center	1	0	0
Right-center	1	1	0
Right	1	1	0

Given that the data are to be presented in crosstab form, the error-cell representation for hypotheses is natural and, we think, quite elegant. Note as well two things. First, we can now give an operational characterization of

---

diversion here. See the discussion in [Hindebrand et al., 1977] of the condensed ordinal form for one example of an alternative crosstab representation.

[6] Or for the cognoscenti of nonmonotonic or defeasible reasoning, "if x then presumably y." But this is a subtlety we defer to another paper.

hypothesis space. If the number of cells in a crosstab representation is C and the number of possible error values (2 in Table 2: 0 for no error and 1 for error) is n, then the number of possible hypotheses is ($n^C - n$). (We subtract n to eliminate the cases in which all cells have the same error value. Presumably, these cannot be interesting predictions.) Thus even for our little example, P_1 is just one of $2^{15} - 2 = 32{,}766$ possible hypotheses for predicting and explaining these data. Second, as we have implied in our first comment just given, it is possible to use more than 2 (0 or 1) error-cell values. Perhaps observations falling in certain cells are intermediate and should have an error value of, say, 0.5. There is nothing in these representations or in Prediction Analysis (see Section 3) that prevents this sort of generalization.

III. PREDICTION ANALYSIS

Put briefly, Prediction Analysis [Hildebrand et al., 1977a,Hildebrand et al., 1977b] is a well-established technique that uses the crosstab and error-cell representations of data and predictions, and also provides a measure of goodness for a prediction (on the given data). We can describe only the basic elements of Prediction Analysis here; much more thorough treatment is available in the open literature. What we find especially intriguing about Prediction Analysis-besides its intuitiveness and its fit with our preferred data representations-are two things. First, it has been shown to subsume most, if not all, standard measures of association for qualitative data, such as Cohen's Kappa, Kendall's τ, and Goodman and Kruskal's gamma (see [Hindebrand et al., 1977a,Hildebrand et al., 1977b] for details). Second, Prediction Analysis was originally motivated to evaluate predictions *ex ante,* for example on the basis of prior theory. But it also can be used *ex post* to select propositions from the data, in which case it is, as one would expect, asymptotically χ^2. Used *ex post,* Prediction Analysis is good for finding “the contenders,” hypotheses that merit careful scientific investigation using standard statistical techniques.

The principal measure of hypothesis value in Prediction Analysis is ∇ (pronounced “dell”), which is defined as follows:

$$\nabla = 1 - \frac{\text{observed error}}{\text{expected error}} \tag{1}$$

Let n_{ij} be the number of observations in cell row i column j, and ω_{ij} be the error value for the cell in row i column j. (Again, although we are holding the discussion in terms of a two-dimensional example, all of this generalizes in a straightforward way.) Then, we may define the observed error for a particular prediction (error-cell table) as

$$\text{observed error} = \sum_{i=1}^{R} \sum_{j=1}^{C} \omega_{ij} \cdot n_{ij} \tag{2}$$

where the number of forms in the row variable is R and the number of forms in the column variable is C.

Finally, the expected error formula is

$$\text{expected error} = \sum_{i=1}^{R} \sum_{j=1}^{C} \omega_{ij} \cdot n_{i\bullet} \cdot n_{\bullet j} / n \tag{3}$$

where

$n_{i\bullet}$ = The number of observations in category i of the first (row) variable

$n_{\bullet j}$ = The number of observations in category j of the second (column) variable

n = The total number of observations

That is, $n_{i\bullet}$ and $n_{\bullet j}$ are the row and column marginals, which are presented in Table 1. Note as well:

1. If the observed error equals 0, then ∇ is 1. This is the highest possible value for ∇.

2. If the observed error equals the expected error, then ∇ is 0. This indicates, roughly, a prediction no better than chance, rather like a correlation of 0. (But remember: standard correlation coefficients apply to real numbers, quantitative variables, not qualitative variables.)

3. ∇ may be negative, arbitrarily so. A negative value is like a negative correlation, but may go lower than –1.

4. In general a higher ∇ indicates a better prediction, but this neglects considerations of parsimony. After all, if all the error cells are set to 0 then ∇ will equal 1.[7] Prediction Analysis uses what it calls the *precision,* which is the expected error rate for a prediction, *P*. Precision in this sense is called U and is defined as

[7] Of course if expected error is 0, the ratio is undefined.

$$U = \sum_{i=1}^{R} \sum_{j=1}^{C} \omega_{ij} \cdot n_{i\bullet} \cdot n_{\bullet j} / (n \cdot n) \qquad (4)$$

Note that if $\omega_{ij} = 0$ for all i, j (i.e., nothing is an error), then U = 0 and if $\omega_{ij} = 1$ for all i, j (i.e., everything is an error), then U = 1.

5. In finding good hypotheses, we seek to maximize ∇. We might think of maximizing ∇ and U jointly, as in $\alpha \cdot \nabla + (1-\alpha) \cdot U$ or in $\nabla \cdot U$;[8] or we might think of U as a constraint on this maximization problem. We might also think of imposing other constraints, such as "naturalness" conditions. For example, in the error cell representation, one might require that there should not be gaps in columns between error and non-error cells. But this is a topic beyond the scope of the present paper. For present purposes, we rely on the user's judgment to impose reasonableness criteria on hypotheses explored.

IV MOTC: A DSS FOR EXPLORING HYPOTHESIS SPACE

MOTC is a prototype implementation of a DSS for exploring hypothesis space. It assumes the two main frameworks we have just discussed (crosstabulation of qualitative data for hypothesis representation, and Prediction Analysis for a measure of goodness for hypotheses) and it meets, or at least addresses, the main requirements we identified above for such a DSS. MOTC is implemented in Visual Basic and Microsoft Access, and runs in a Windows environment.

The central, dominating metaphor in MOTC is the representation of variables (dimensions) as binned bars. A single bar corresponds to a single

[8] $\nabla \bullet U = U - K$ or the absolute reduction in error of the prediction. One might instead, e.g., prefer to use the relative reduction in error.

variable. Bars are arrayed horizontally, and are divided by vertical lines indicating bins. Each bin corresponds to a category for the variable in question. Thus, in our previous example the bar for *Party Affiliation* would have three bins, while the bar for *Support* would have five bins. A user may right-click on a bar and MOTC will present information about the underlying binning arrangement. See the figures in Section 5 for illustration. The width of a bin as displayed represents the percentage of records in the relevant data set that have values falling into the bin in question. Wider bins indicate proportionately larger numbers of records. MOTC as presently implemented allows up to eight variables to be represented as bars on the display. A bar may have any number of bins. This is in fact an interesting and nontrivial degree of multidimensionality (and see our discussion in Section 6 of the focus+context technique used by Rao and Card in their Table Lens program [Rao and Card, 1994]).

MOTC as currently implemented has two modes of operation: hypothesis hunting (also known as: brush) mode, and hypothesis evaluation (also known as: prediction) mode. In hypothesis hunting mode, users use brushing with the mouse to display relationships among variables. Users choose particular bins and brush them with a chosen color by clicking on them. MOTC responds by applying the same color to bins associated with other variables. For example, if the user brushes bin 3 of variable 1 with purple, MOTC might respond by covering 25% of bin 2 of variable 4 in purple, indicating thereby that 25% of the records associated with bin 2 of variable 4 also are associated with bin 3 of variable 1. (See the various figures that follow for illustrations.) A user may brush more than one bin with a single color, either within or without a single variable. The effect is a logical “or” for bins within a single variable (bar) and an “and” for bins in different variables. Further, suppose purple is used to brush bins 1 and 2 of variable X, bins 4 and 5 of variable Y, and bins 7 and 8 of variable Z. Suppose further that we are in prediction mode (see below) and that we want X and Y to predict Z. Then, the equivalent representation in Prediction Analysis terminology is:

$((X_1 \vee X_2) \wedge (Y_4 \vee Y_5)) \sim> (Z_7 \vee Z_8)$

MOTC presently supports up to five colors for brushing. Each color used corresponds to a separate ~> rule in terms of Prediction Analysis. Working in brush mode, the user explores hypothesis space, with MOTC providing feedback by coloring bins in the unbrushed bars (predicted variables). The user thus gets a rough idea of where the "big hits" in the predictions lie.

In hypothesis evaluation, or prediction, mode the user brushes-clicks and colors-bins in the predictor *and* predicted variable bars. In essence, the user is interactively populating a higher-dimensional version (up to 8 dimensions in the current implementation) of an error-cell table, as in Table 2. Doing so specifies a hypothesis and MOTC responds by calculating and displaying ∇ and U for the hypothesis.

Working iteratively, the user may explore hypothesis space by switching back and forth between hypothesis hunting mode and hypothesis evaluation mode. This continues until the user reaches reflective equilibrium.

V. MOTC AT WORK

For MOTC—or any similar program—to be evaluated as an effective tool for KDD, it must be shown that interesting patterns, trends, or facts about the data can be discovered easily and quickly. Our experience with MOTC—admittedly preliminary and biased—is quite favorable in this regard. Extensive empirical investigation will be required to determine how best to design a MOTC-like tool and how well it can work. That remains for the future. Our purpose in this section is to present three cases in which MOTC is used to good effect.

The first case, in Section 5.1, describes the use of MOTC in some detail. Our aim here is to indicate MOTC's core features and to give the reader a feel for what an analysis effort with MOTC is like. The data analyzed, DataSet1, were privately generated by one of the authors, using distributions of random numbers

and adding a fair amount of noise. DataSet1 is a test dataset to see if an analyst can recover the rules privately (and with noise) placed in the data. This is a preliminary test for MOTC, but a realistic enough one even so. In this particular test, as well as in others we have run the analysts have indeed done quite well. Here, we are making no empirical claim, other than a *prima facie* one; our purpose is simply to show MOTC at work and let the reader form a judgment on that.

We are briefer in presenting our second and third cases. Our aim here is to show MOTC at work in two useful contexts. The first of these (our second case, in Section 5.2) presents an analyst's discussion of an initial exploration of data returned from a survey of information systems professionals. As is appropriate for MOTC, our emphasis is on hypothesis generation, not testing. Our third case, in Section 5.3, describes MOTC in action for exploring decision surfaces-collections of solutions presented by optimization solvers-for post-solution analysis and decision making, a process called *candle-lighting analysis* [Branley et al., 1997,Kimbrough et al., 1993]. Together, these cases hint at the considerable scope of potential application for a system such as MOTC.

5.1 MOTC ILLUSTRATED

Our purpose in this section is to walk the reader through a sequence of hypotheses analyses using the MOTC tool. In this example the first action to be performed is loading the Set_1 database into MOTC. Figure 1 shows the data configuration dialog box for this data set.

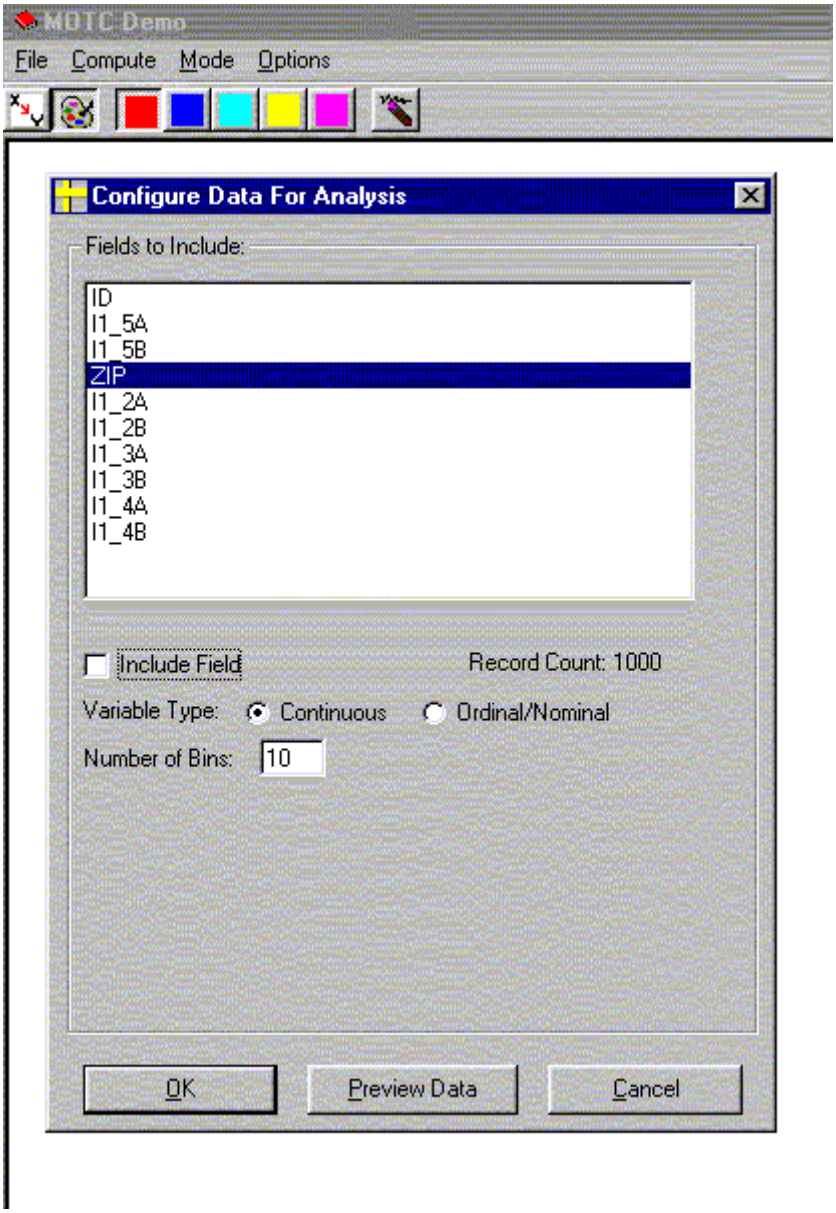
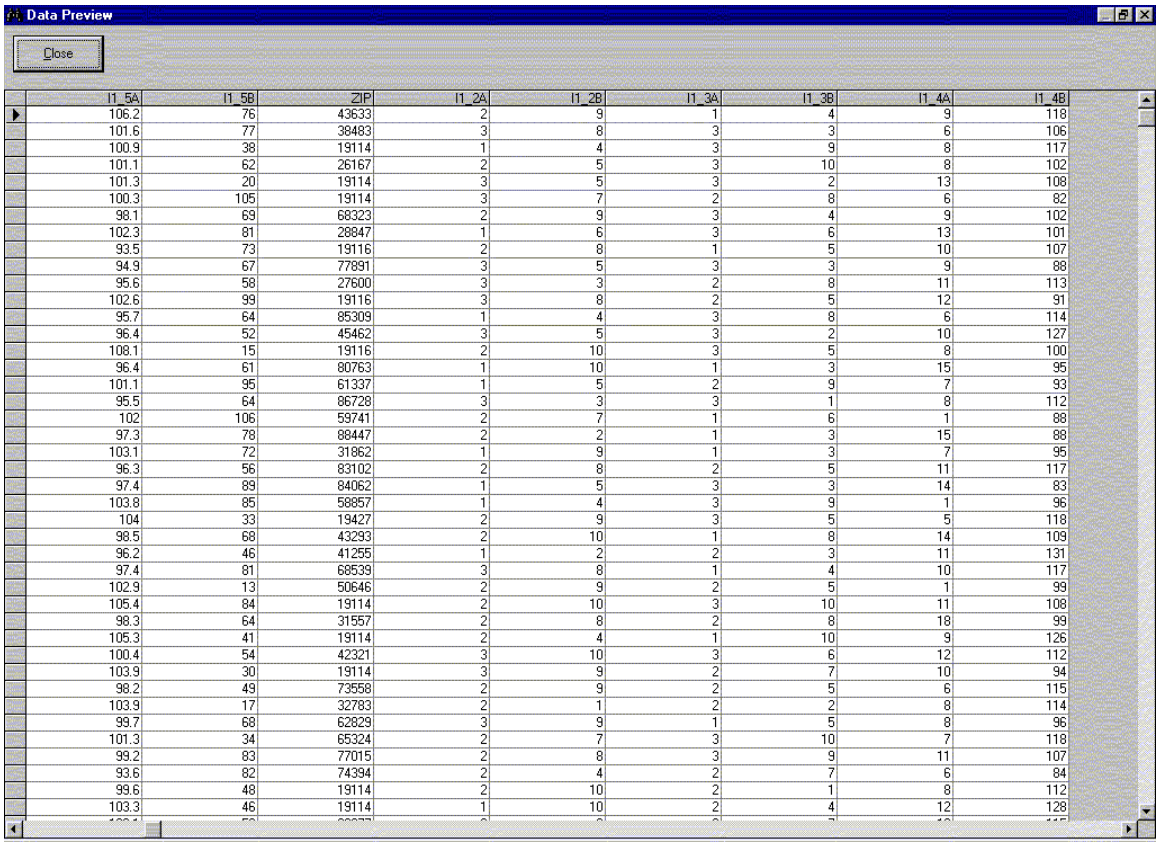


Figure 1. MOTC Data Configuration Dialog Box for the Set_1 Database

To get a good understanding of the type of data in the data set at a glance, one uses the “Preview Data” button to display the contents of the database. Outside of the field ZIP representing zip codes, it is difficult to determine just by looking at the field names what each field represents. Looking at the values of some of the fields it seems as though some continuous fields may actually be ordinals. This

seems to be the case for fields I1_2A, I1_2B, I1_3A, and I1_3B. Figure 2 shows a preview of the Set_1 database.



I1_5A	I1_5B	ZIP	I1_2A	I1_2B	I1_3A	I1_3B	I1_4A	I1_4B
106.2	76	43633	2	9	1	4	9	118
101.6	77	38483	3	8	3	3	6	106
100.9	38	19114	1	4	3	9	8	117
101.1	62	26167	2	5	3	10	8	102
101.3	20	19114	3	5	3	2	13	108
100.3	105	19114	3	7	2	8	6	82
98.1	69	68323	2	9	3	4	9	102
102.3	81	28847	1	6	3	6	13	101
93.5	73	19116	2	8	1	5	10	107
94.9	67	77891	3	5	3	3	9	88
95.6	58	27600	3	3	2	8	11	113
102.6	99	19116	3	8	2	5	12	91
95.7	64	85309	1	4	3	8	6	114
96.4	52	45462	3	5	3	2	10	127
108.1	15	19116	2	10	3	5	8	100
96.4	61	80763	1	10	1	3	15	95
101.1	95	61337	1	5	2	9	7	93
95.5	64	86728	3	3	3	1	8	112
102	106	59741	2	7	1	6	1	88
97.3	78	88447	2	2	1	3	15	88
103.1	72	31862	1	9	1	3	7	95
96.3	56	83102	2	8	2	5	11	117
97.4	89	84062	1	5	3	3	14	83
103.8	85	58857	1	4	3	9	1	96
104	33	19427	2	9	3	5	5	118
98.5	68	43293	2	10	1	8	14	109
96.2	46	41255	1	2	2	3	11	131
97.4	81	68539	3	8	1	4	10	117
102.9	13	50646	2	9	2	5	1	99
105.4	84	19114	2	10	3	10	11	108
98.3	64	31557	2	8	2	8	18	99
105.3	41	19114	2	4	1	10	9	126
100.4	54	42321	3	10	3	6	12	112
103.9	30	19114	3	9	2	7	10	94
98.2	49	73558	2	9	2	5	6	115
103.9	17	32783	2	1	2	2	8	114
99.7	68	62829	3	9	1	5	8	96
101.3	34	65324	2	7	3	10	7	118
99.2	83	77015	2	8	3	9	11	107
93.6	82	74394	2	4	2	7	6	84
99.6	48	19114	2	10	2	1	8	112
103.3	46	19114	1	10	2	4	12	128

Figure 2. Preview of the Set_1 Database

After previewing the data, you can determine whether you want the fields to be viewed as continuous or ordinal/nominal by selecting the “Ordinal/Nominal” radio button. This is done for I1_2A with the resulting window shown in Figure 3. Since there are only three values, 1, 2, and 3, then this field is best treated as an ordinal and not a continuous number. To do this keep the “Ordinal/Nominal” radio button set for I1_2A. This is also true for I1_2B (integers from 1 to 10), I1_3A (integers from 1 to 3), and I1_3B (integers from 1 to 10), so each of these fields are also changed from continuous to ordinal.

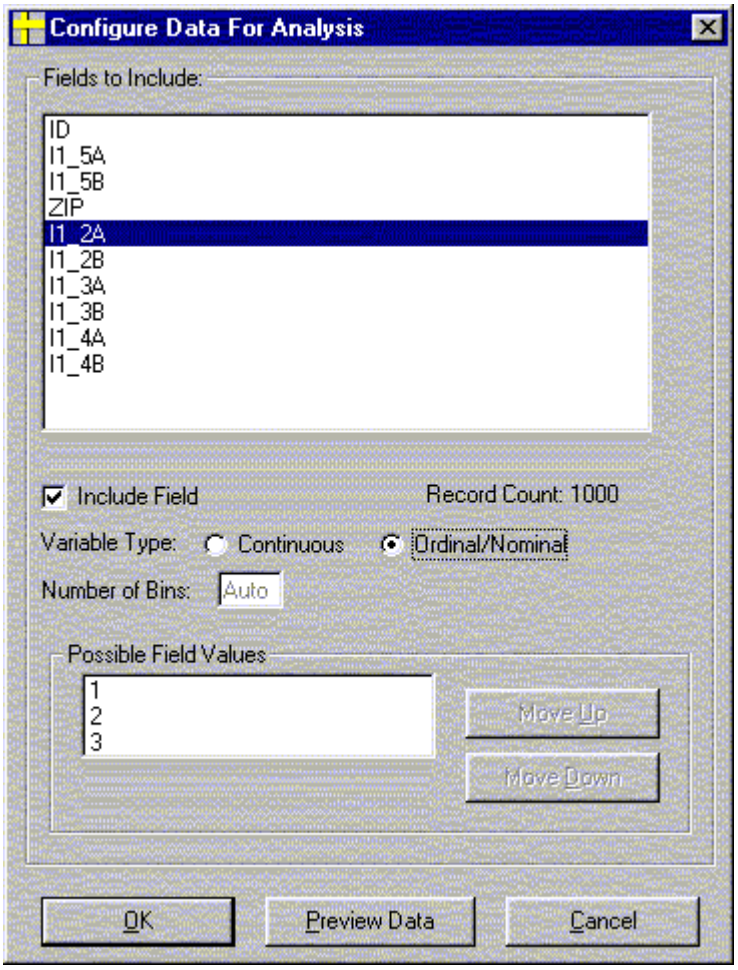


Figure 3. Setting the I1_2A Variable to Type Ordinal

After changing those four fields from continuous to ordinal, the ID and ZIP fields are deselected from inclusion in the analysis because the current implementation of MOTC is limited to a maximum of eight fields. The choice of ID is obvious. ZIP is not included in the analysis now, but will be analyzed latter in this session. Clicking OK in the dialog box results in the loading of the selected fields from the Set_1 database into MOTC and the display shown in Figure 4.

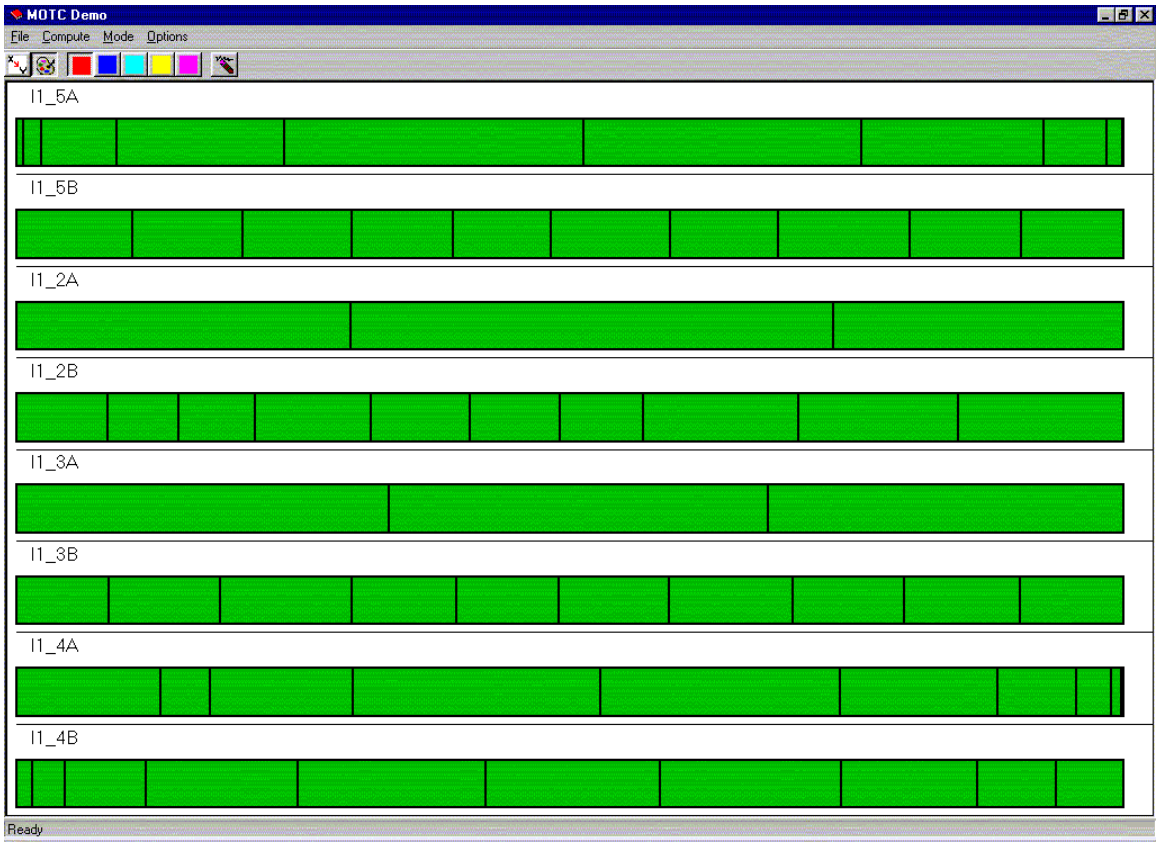


Figure 4. Initial MOTC Analysis Display Of The Eight Selected Variables

At this point hypothesis formulation can begin by selecting different bins within a field and observing the effects on bin populations in other fields. In this case, without knowing any understandable field names, this process is performed somewhat randomly. After selecting and deselecting bins over a period of time, a relationship seems to have emerged between some bins in field I1_5B and I1_4B as shown in Figure 5.

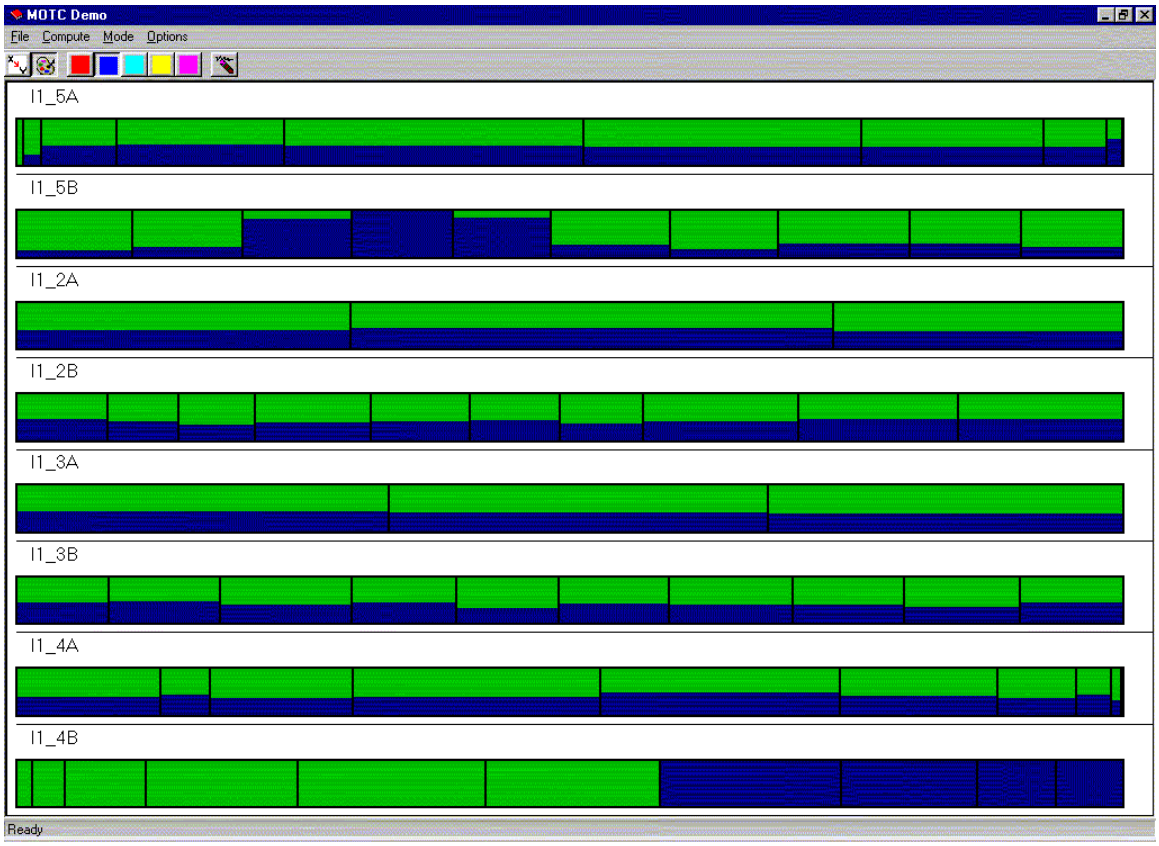


Figure 5. Hypothesis Hunting With Bins 7 to 10 of Variable I1_4B Selected

At this point when a possible hypothesis seems to be found with just a couple of the fields involved, it is best that a new MOTC application be invoked with only those two fields included for data analysis. In this case the same set_1 database is loaded with only the i1_5b and i1_4b fields selected for analysis. Once the two fields are displayed, the appropriate bins of both fields are selected in prediction mode of the tool. Once this is done, the predicted variable must be set. In this case, the i1_4b field is selected as the predicted variable and is highlighted. The results of performing the above actions are shown in figure 6.

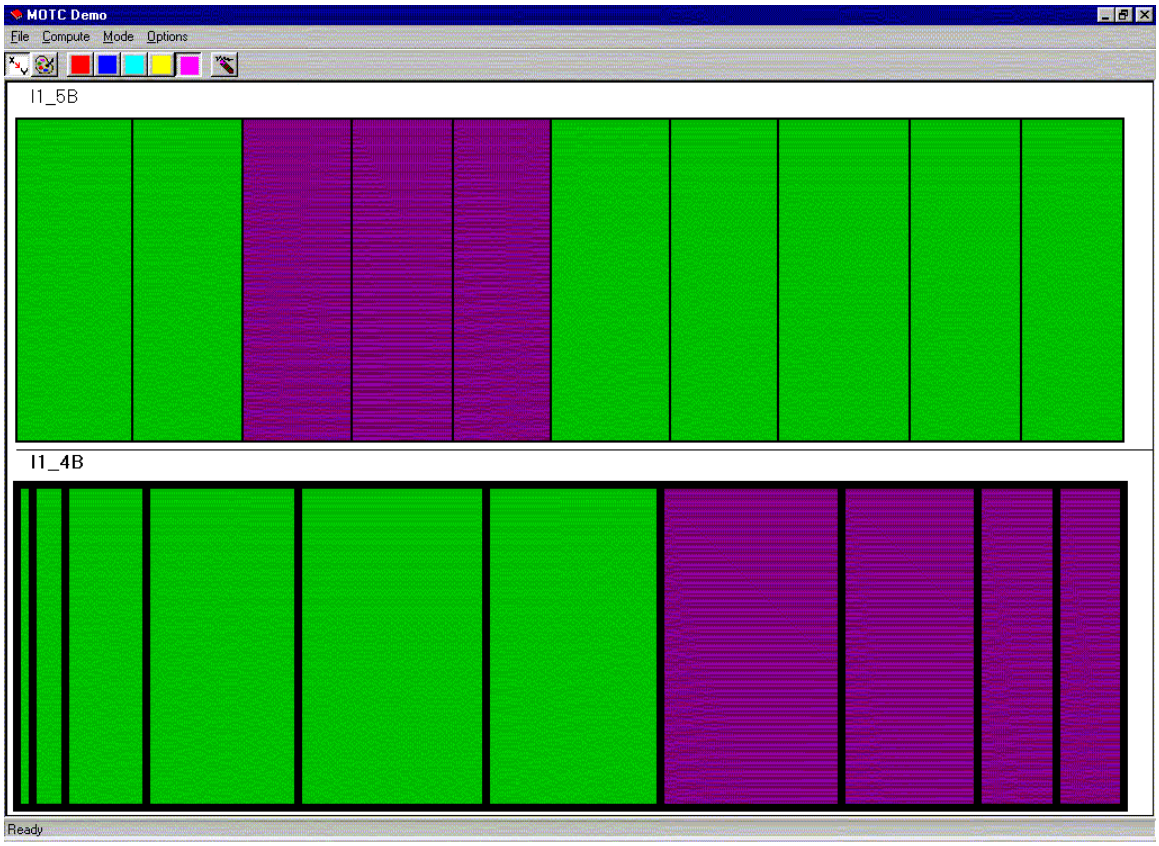


Figure 6. Hypothesis Declaration, Bins 3 to 5 of I1_5B ane Bins 7 to 10 of I1_4B

To check this hypothesis, the ∇ value and Precision must be calculated. The resultant values for this hypothesis are shown in Figure 7. From experience, this seems to be a valid hypothesis for the data set since the ∇ value is high with an acceptable precision.

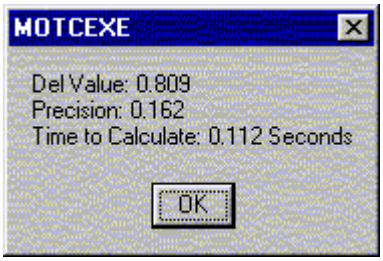


Figure 7. Calculated ∇ (Del) Value and Precision for the I1_5B & I1_4B Hypothesis

We can now make-or at least conjecture-a more formal statement about the relationship between variables I1_5B and I1_4B using Prediction analysis nomenclature. From this analysis it can be said “If bins 3, 4, and 5 of I1_5B, then predict bins 7, 8, 9, and 10 of I1_4B.” This does not mean much unless the bins can be mapped to values. So the next step is to display the data bins to determine this mapping. Figures 8 and 9 are the bin data displays for both the I1_5B and I1_4B variables. From this it can be seen that bins 3, 4, and 5 of I1_5B map to the range 30 to 60. Also, bins 7 through 10 of I1_4B map to values greater than 107.6. Therefore this hypothesis can be stated more specifically as “If I1_5B is between 30 and 60, then it can be predicted that I1_4B is greater than 107.6.” This is the first data prediction.

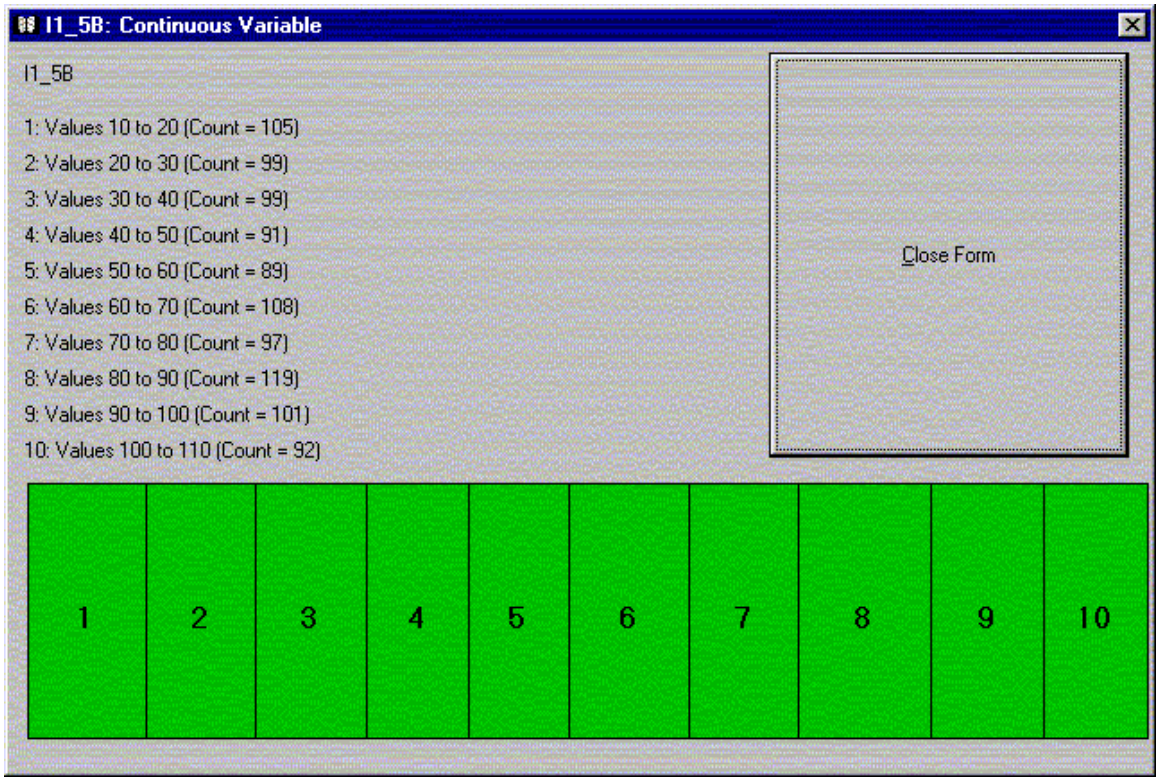


Figure 8. Bin Data Display for the Variable I1_5B

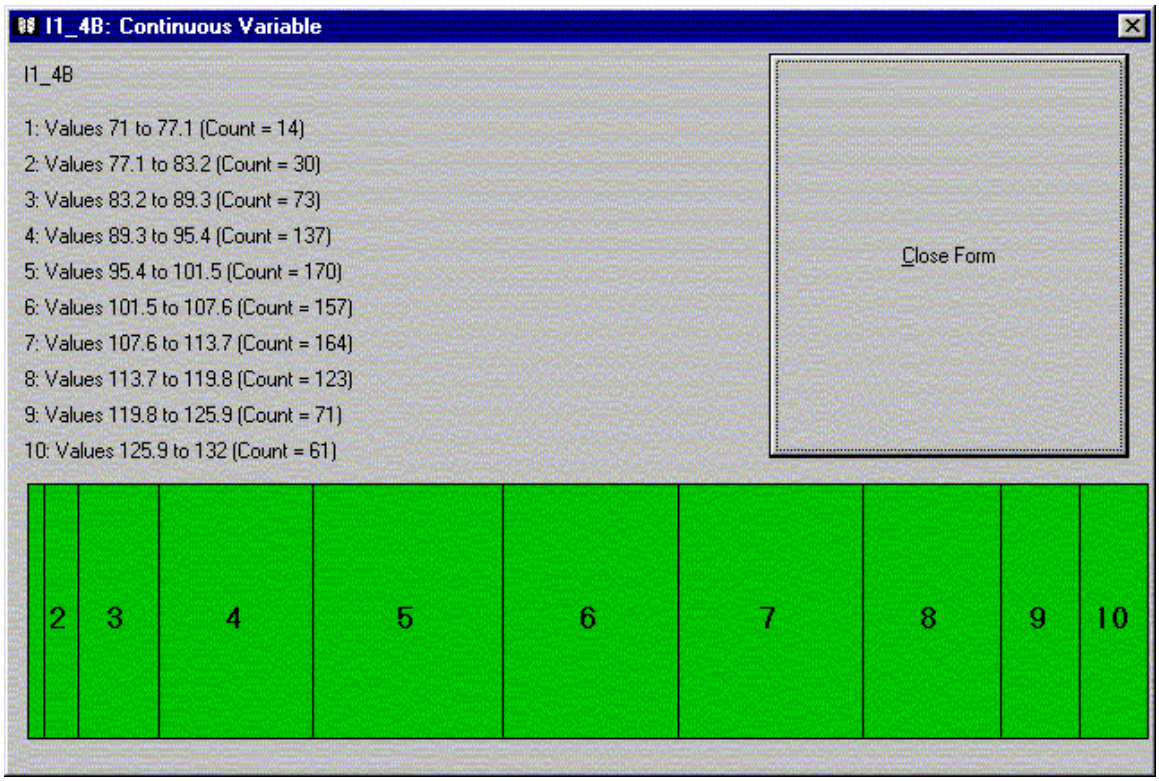


Figure 9. Bin Data Display for the Variable I1_4B

Now we will go back to the original MOTC application and search for more relationships. Again, after selecting and deselecting bins over a period of time, a relationship seems to have emerged between some bins in field I1_3B and I1_2B as shown in Figure 10.

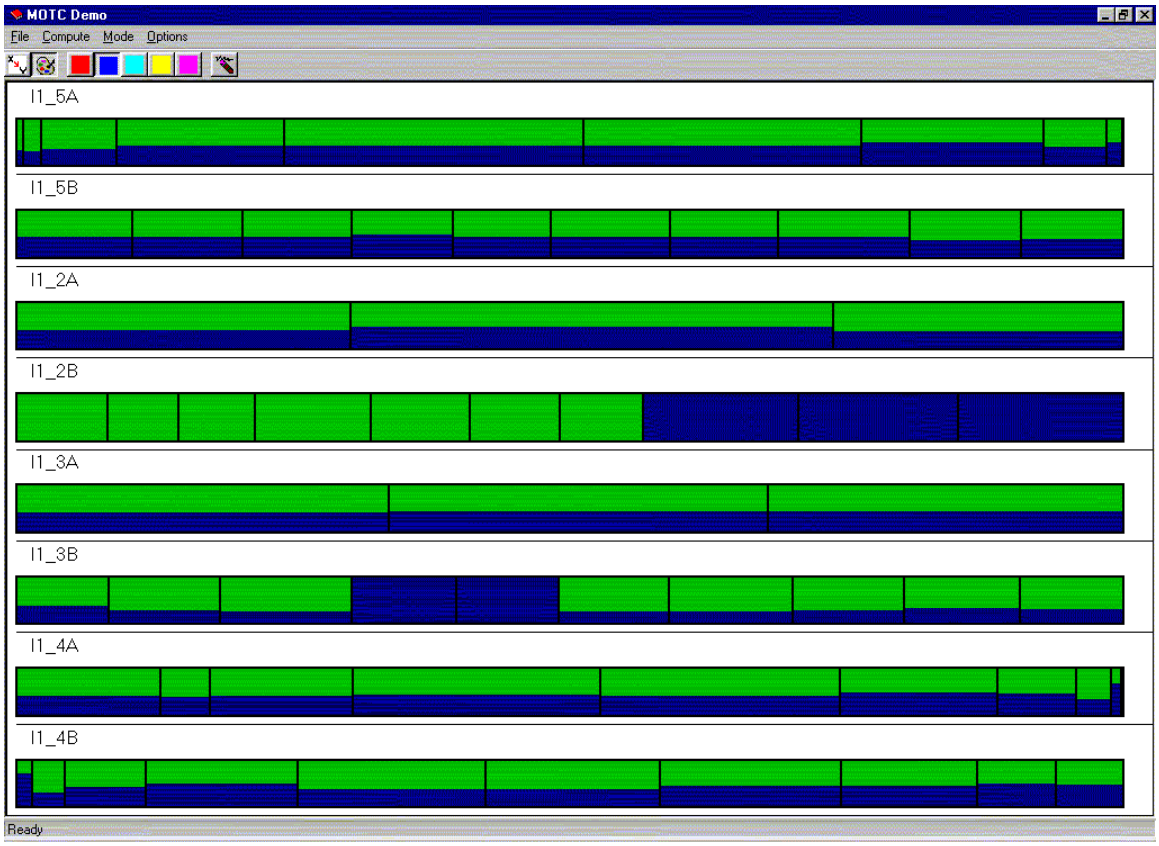


Figure 10. Hypothesis Hunting with Bins 8 to 10 of Variable I1_2B Selected

Again a new MOTC application is invoked with only those two fields included for data analysis. In this case the same Set_1 database is loaded with only the I1_3B and I1_2B fields selected for analysis. Once the two fields are displayed, the appropriate bins of both fields are selected in prediction mode of the tool. Once this is done, the predicted variable must be set. In this case, the I1_2B field is selected as the predicted variable and is highlighted. The results of performing the above actions are shown in Figure 11.

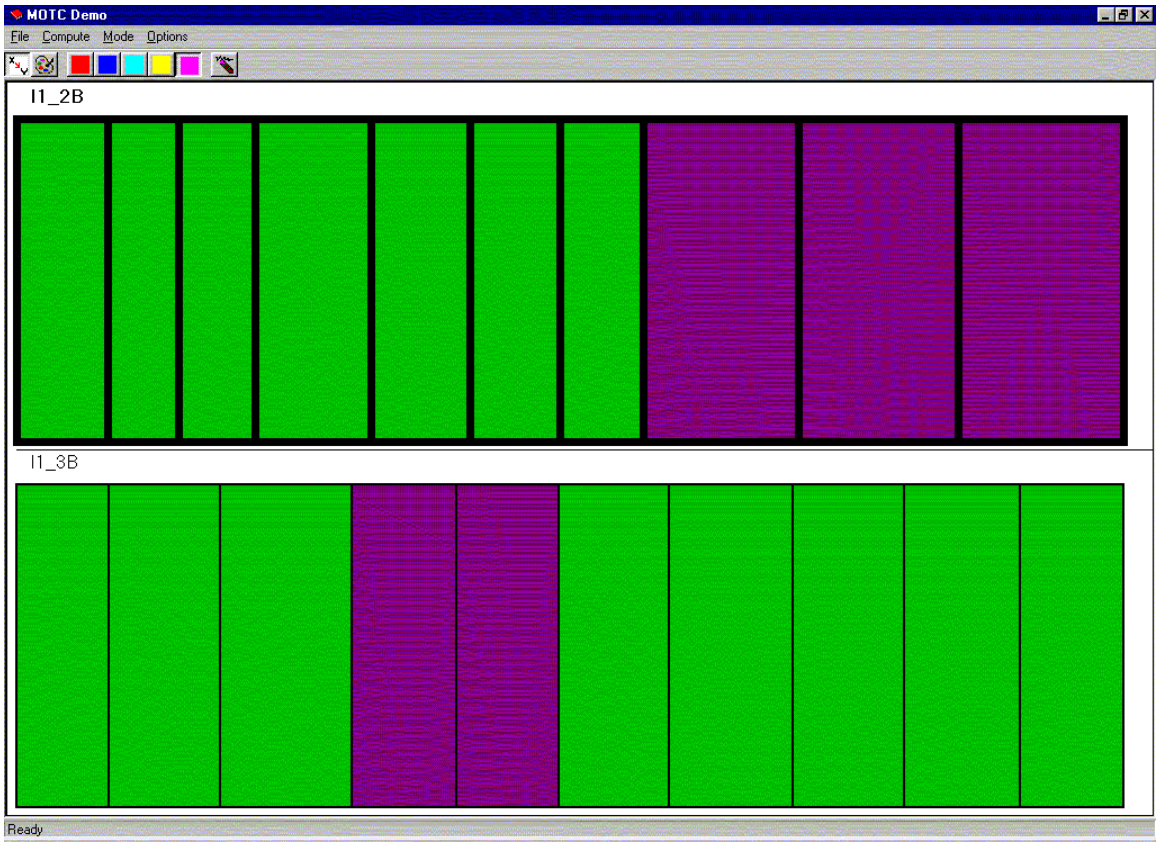


Figure 11. Hypothesis Declaration, Bins 4 and 5 of I1_3B & Bins 8 to 10 of I1_2B

To check this hypothesis, the ∇ value and Precision must be calculated. The resultant values for this hypothesis are shownin figure 12. This seems to be a valid hypothesis for the data set since the ∇ value is high with an acceptable, though low, precision.

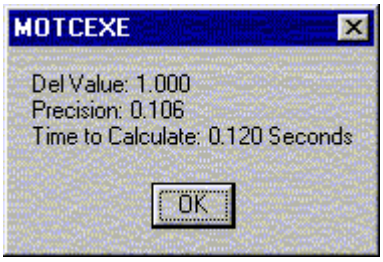


Figure 12. Calculated ∇ (Del) Value and Precision for the I1_3B & I1_2B Hypothesis

We can now make a more formal statement about the relationship between variables I1_3B and I1_2B using Prediction analysis nomenclature. From this analysis it can be said “If bins 4 and 5 of I1_3B, then predict bins 8, 9, and 10 of I1_2B.” This does not mean much unless the bins can be mapped to values. So the next step is to display the data bins to determine this mapping. Figures 13 and 14 are the bin data displays for both the I1_3B and I1_2B variables. From this it can be seen that bins 4 and 5 of I1_3B map to the values 4 and 5 because I1_3B is an ordinal field. Also, bins 8 through 10 of I1_2B map to values the values 8, 9, and 10 respectively, since again I1_2B is an ordinal field. Therefore this hypothesis can be stated more specifically as “If I1_3B is either 4 or 5, then it can be predicted that I1_2B is either 8, 9, or 10.” This is the second data prediction.

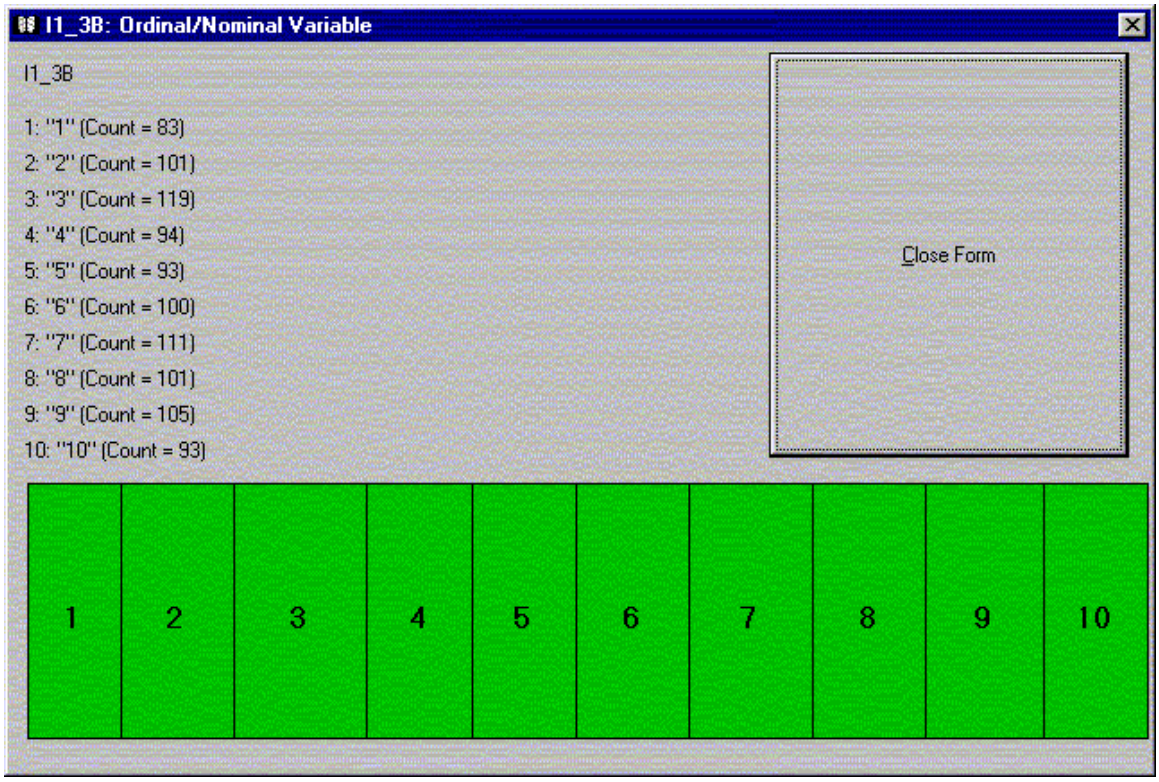


Figure 13. Bin Data Display for the Variable I1_3B

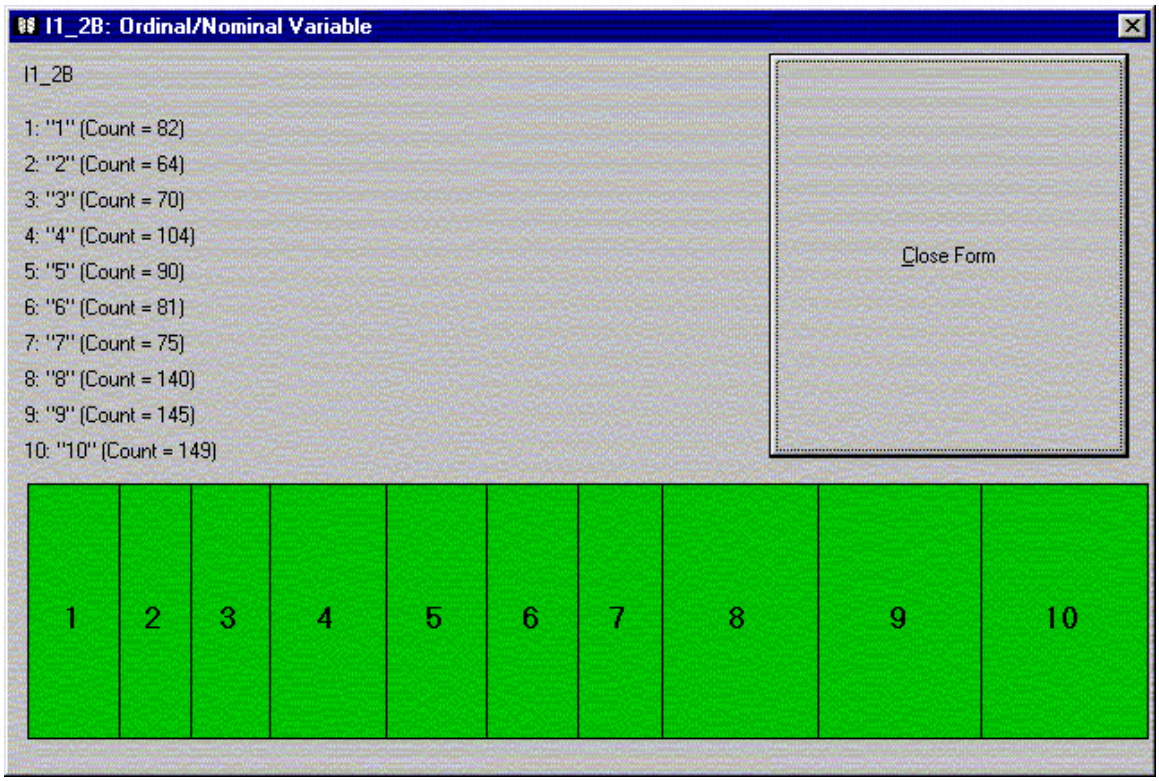


Figure 14. Bin Data Display for the Variable I1_2B

Yet again, we will go back to the original MOTC application and search for more relationships. After selecting and deselecting bins over a period of time, a relationship seems to have emerged between some bins in field I1_5A and I1_2A as shown in Figure 15.

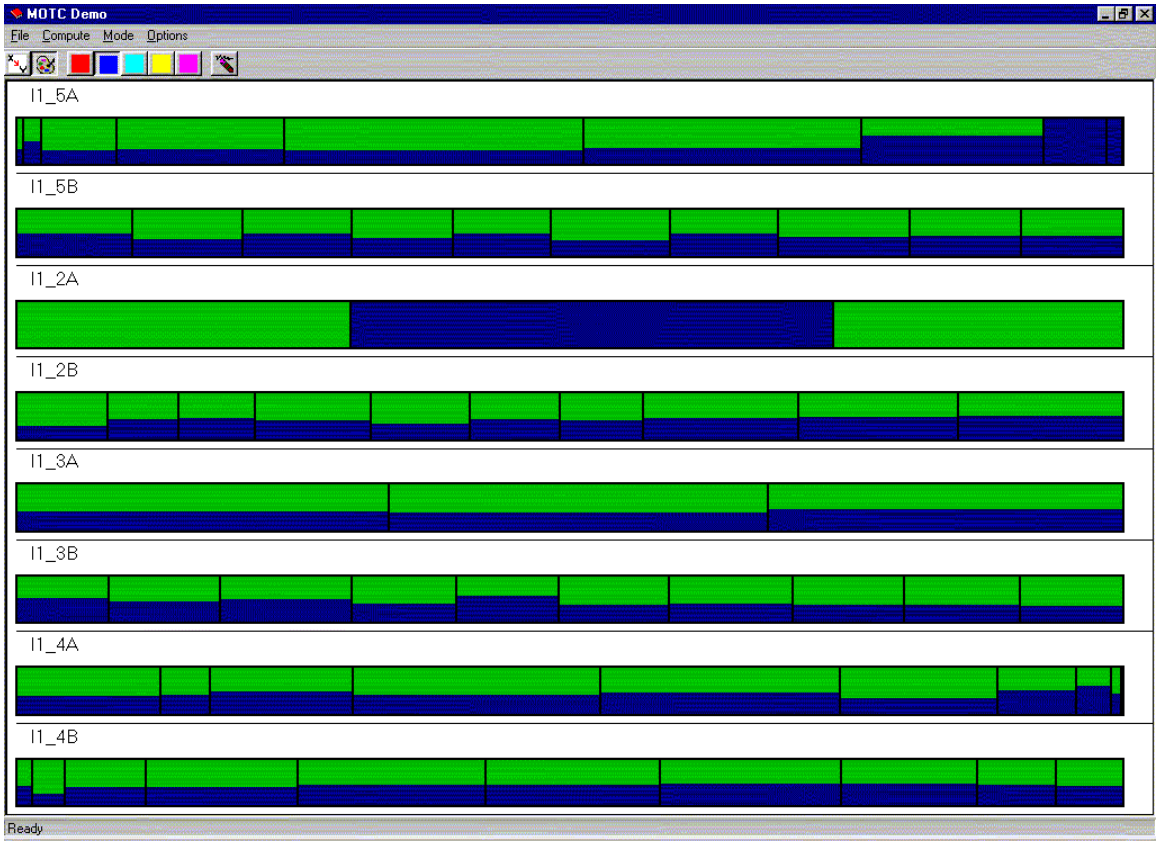


Figure 15. Hypothesis Hunting with Bin 2 of Variable I1_2A selected

Again a new MOTC application is invoked with only those two fields included for data analysis. In this case the same Set_1 database is loaded with only the I1_5A and I1_2A fields selected for analysis. Once the two fields are displayed, the appropriate bins of both fields are selected in prediction mode of the tool. Once this is done, the predicted variable must be set. In this case, the I1_2A field is selected as the predicted variable and is highlighted. The results of performing the above actions are shown in Figure 16.

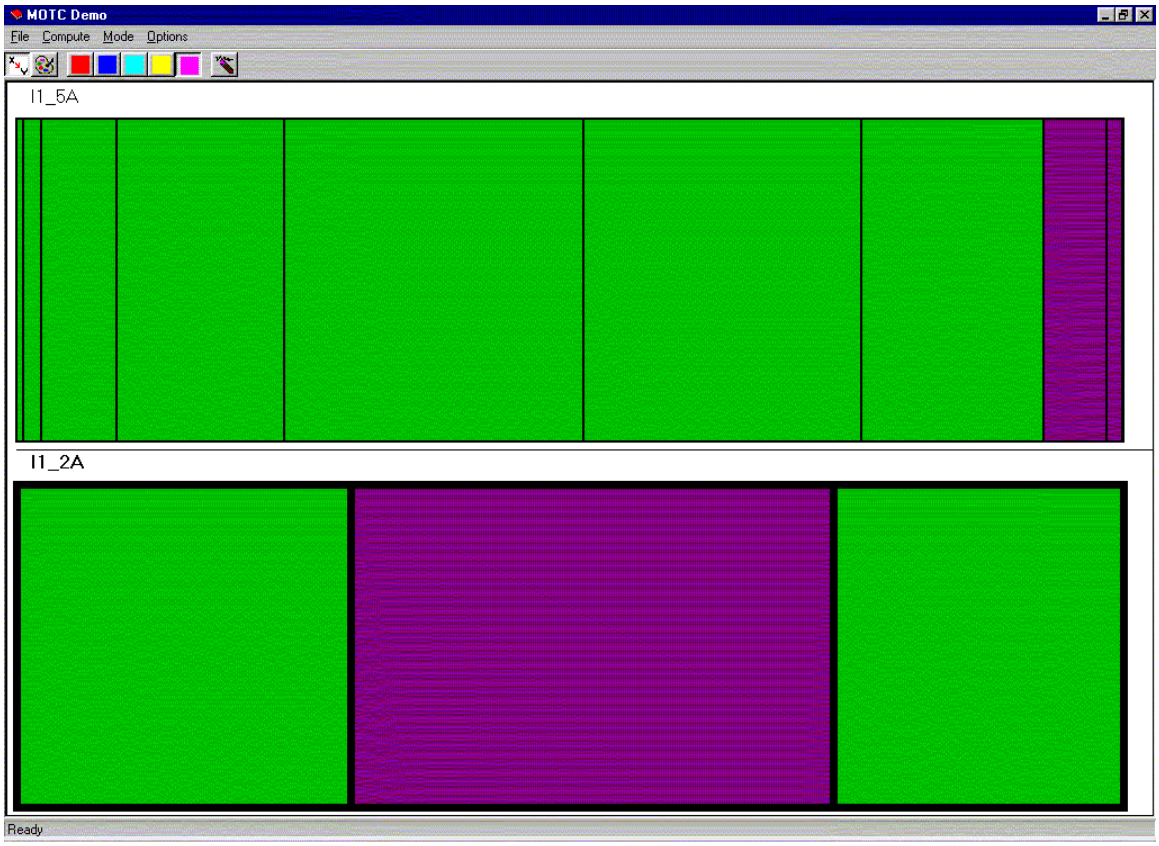


Figure 16. Hypothesis Declaration, bins 8 to 10 of I1_5A & bin 2 of I1_2A

To check this hypothesis, the ∇ value and Precision must be calculated. The resultant values for this hypothesis are shown in Figure 17. This seems to be a valid hypothesis for the data set since the ∇ value is high with an acceptable, though again low, precision.

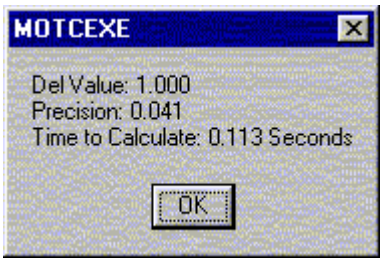


Figure 17. Calculated ∇ (Del) value and Precision for the I1_5A & I1_2A hypothesis

We can now make a more formal statement about the relationship between variables I1_5A and I1_2A using Prediction analysis nomenclature. From this analysis it can be said “If bins 9 and 10 of I1_5A, then predict bin 2 of I1_2A.” This does not mean much unless the bins can be mapped to values. So the next step is to display the data bins to determine this mapping. Figures 18 and 19, are the bin data displays for both the I1_5A and I1_2A variables. From this it can be seen that bins 9 and 10 of I1_5A map to the values greater than 106.76. Also, bin 2 of I1_2A maps to the value 2, since I1_2A is an ordinal field. Therefore this hypothesis can be stated more specifically as “If I1_5A is greater than 106.76, then it can be predicted that I1_2A is 2.” This is the third data prediction.

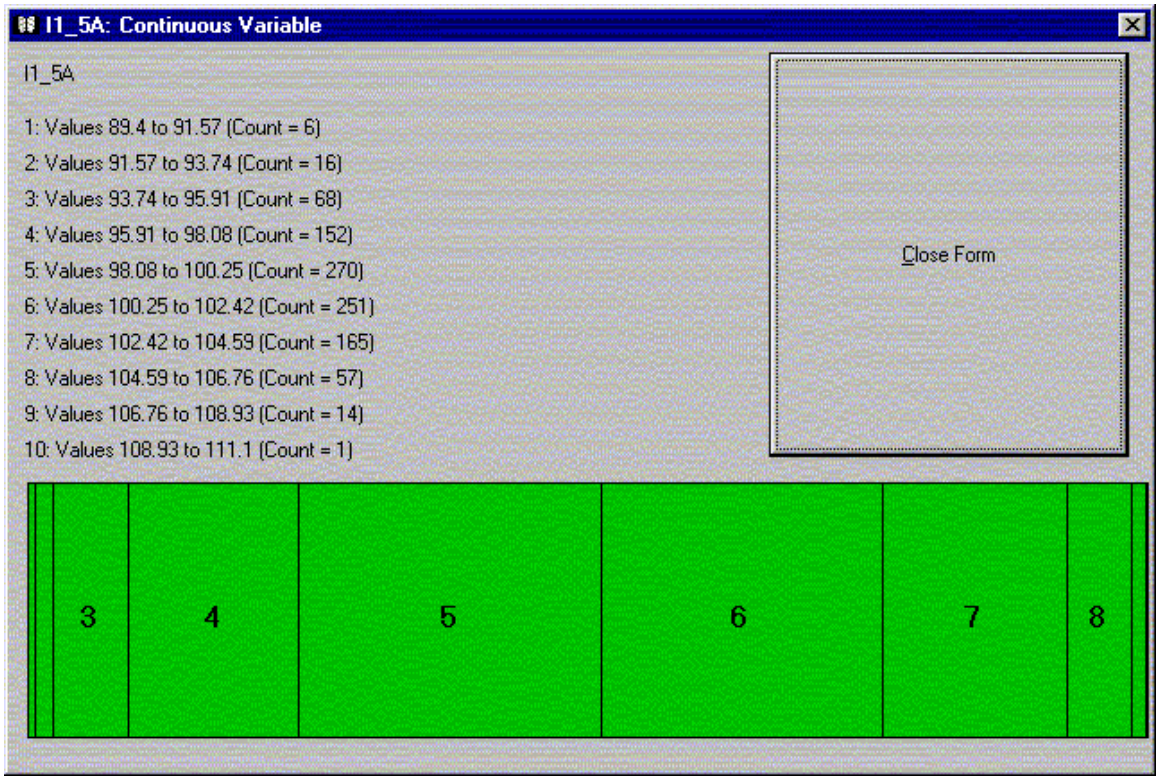


Figure 18. Bin Data Display for the Variable I1_5A

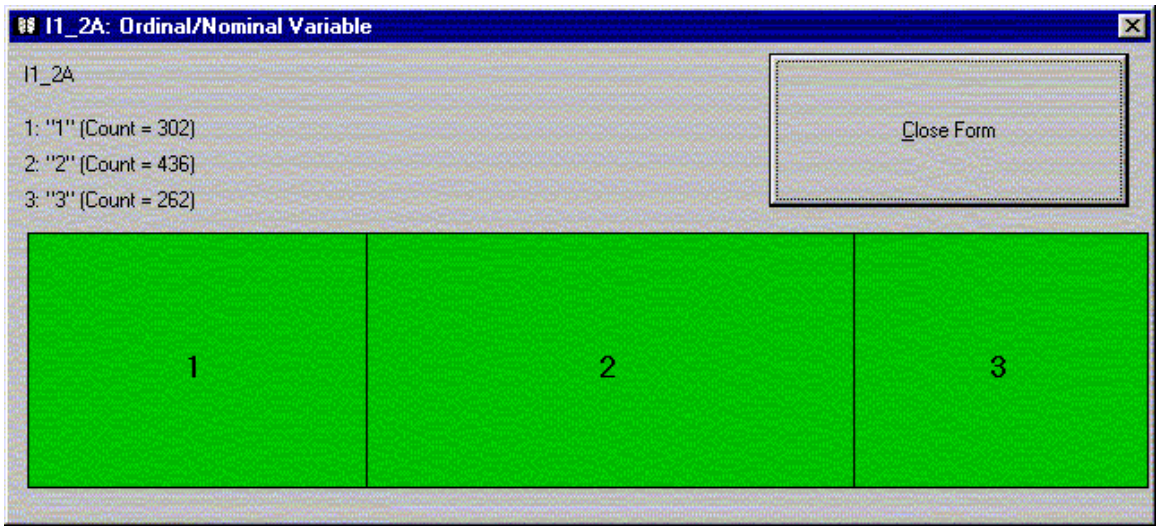


Figure 19. Bin Data Display for the Variable I1_2A

Finally, we will go back to the original MOTC application and search for more relationships this time including the ZIP field from the database. To do this, first the Set_1 database is loaded into MOTC with the ID, I1_5A, and I1_5B variables excluded and the I1_2A, I1_2B, I1_3A, and I1_3B fields changed from continuous to ordinal, as before. After selecting and deselecting bins over a period of time, a relationship seems to have emerged between some bins in field I1_4A and ZIP as shown in Figure 20.

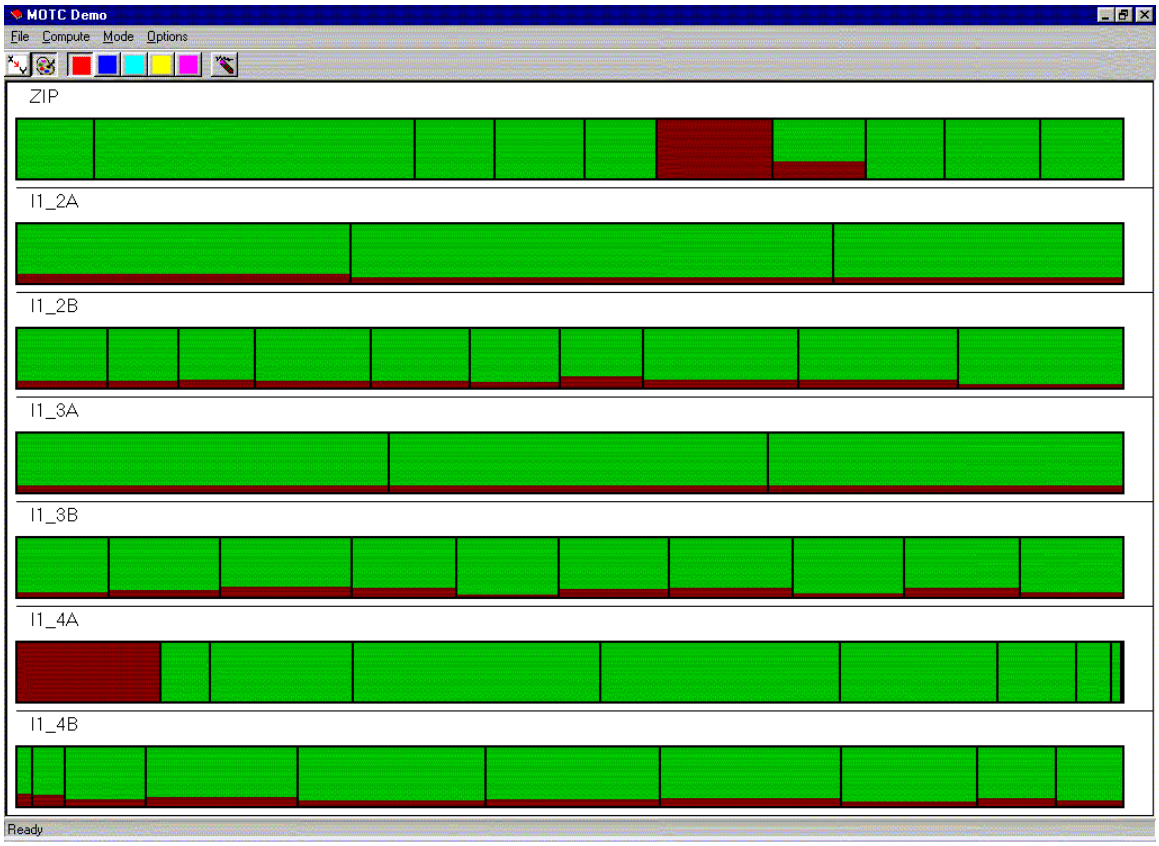


Figure 20. Hypothesis Hunting with Bin 1 of Variable I1_4A Selected

Again a new MOTC application is invoked with only those two fields included for data analysis. In this case the same Set_1 database is loaded with only the I1_4A and ZIP fields selected for analysis. Once the two fields are displayed, the appropriate bins of both fields are selected in prediction mode of the tool. Once this is done, the predicted variable must be set. In this case, the ZIP field is selected as the predicted variable and is highlighted. The results of performing the above actions are shown in Figure 21.

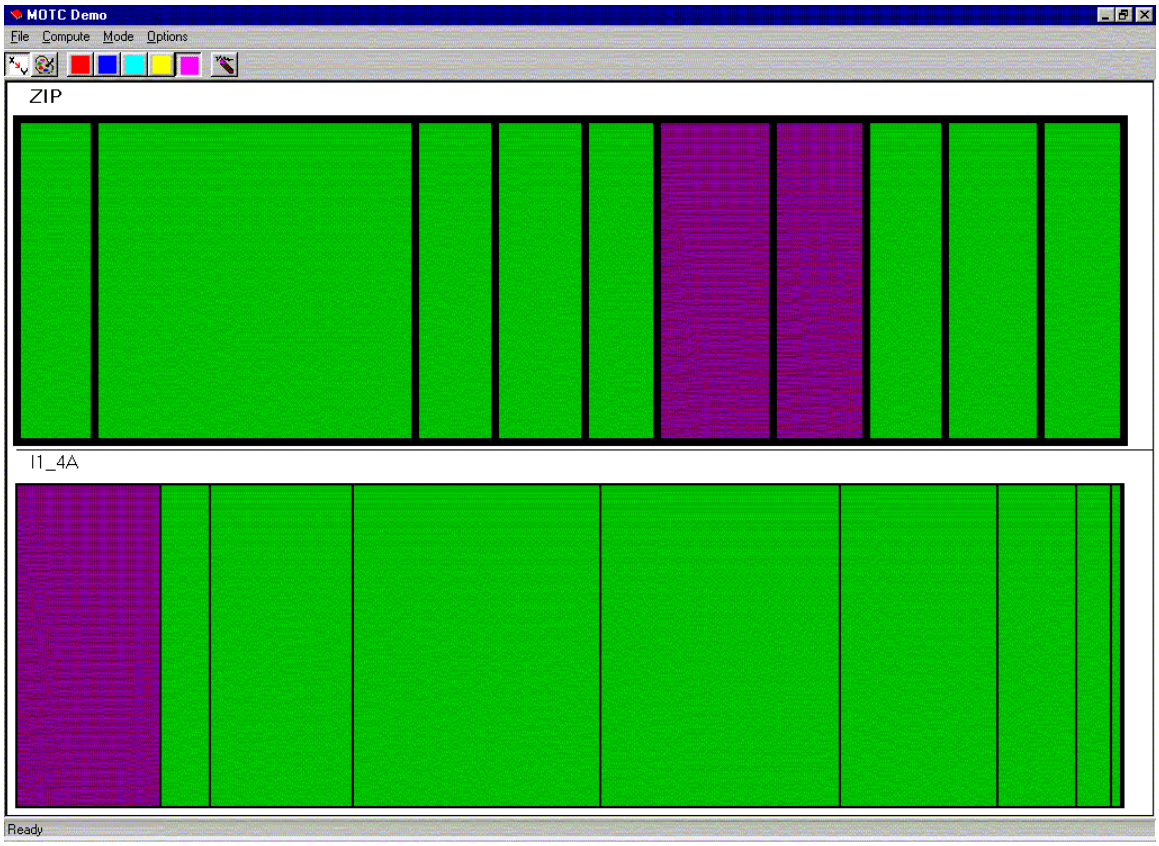


Figure 21. Hypothesis Declaration, Bin 1 of I1_4A and Bins 6 and 7 of ZIP

To check this hypothesis, the ∇ value and Precision must be calculated. The resultant values for this hypothesis are shown in Figure 22. This seems to be a valid hypothesis for the data set since the ∇ value is high with an acceptable, though again low, precision.

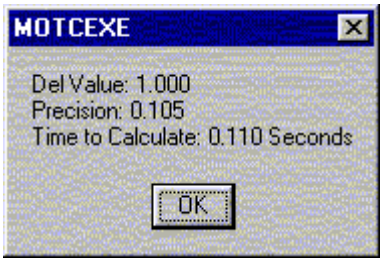


Figure 22. Calculated ∇ (Del) Value and Precision for the I1_4A & ZIP Hypothesis

We can now make a more formal statement about the relationship between variables I1_4A and ZIP using Prediction analysis nomenclature. From this analysis it can be said “If bin 1 of I1_4A, then predict bins 6 and 7 of ZIP.” This does not mean much unless the bins can be mapped to values. So the next step is to display the data bins to determine this mapping. Figures 23 and 24, are the bin data displays for both the I1_4A and ZIP variables. From this it can be seen that bin 1 of I1_4A maps to the values 1 through 3. Also, bins 6 and 7 of ZIP map to zip codes between 55045 and 66010. Therefore this hypothesis can be stated more specifically as “If I1_4A is between 1 and 3 inclusive, then it can be predicted that the zip code is between 55045 and 66010.” This is the fourth and final data prediction.

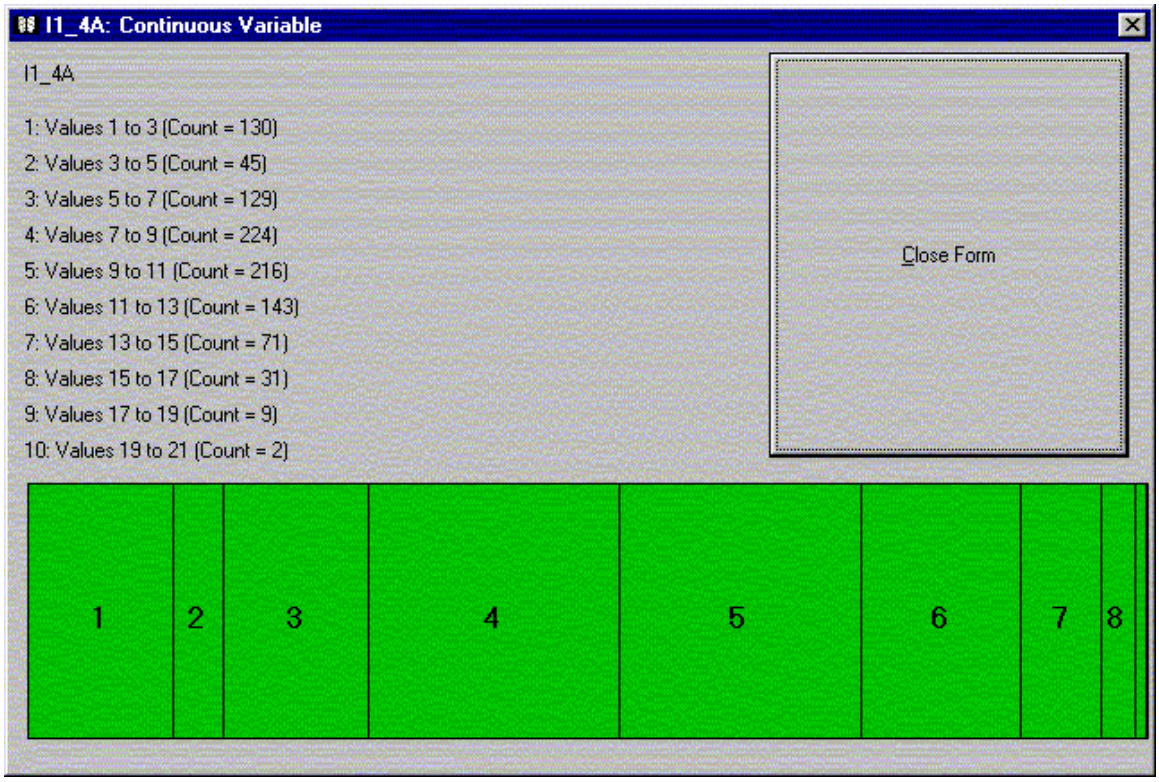


Figure 23. Bin Data Display for the Variable I1_4A

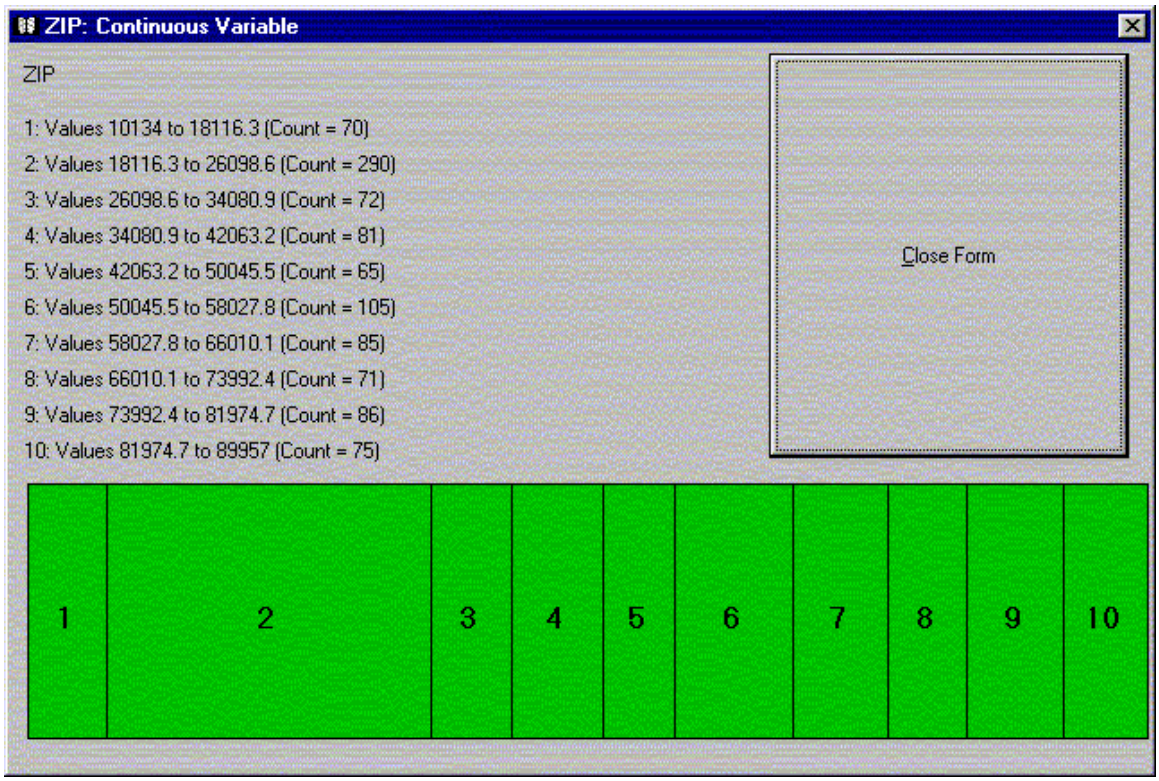


Figure 24. Bin Data Display for the Variable ZIP

This concludes the Data Set 1 example for using MOTC.

5.2 ANALYSIS OF SURVEY DATA

This section describes using MOTC on data from a survey that was taken to determine software workers' opinion of quality. What interested us about this dataset is that survey data, in general, is categorical-mostly nominal-in nature. This is especially true in this survey: of the 40+ questions, all but two were nominally scaled. Typically, in analyzing categorical survey data one engages in seemingly endless generation of cross-tabulated tables; however, with MOTC, the cross-tabulated tables can be generated very easily using any dimensions and constraints one wishes.

The data analyzed were from a software engineering survey on quality issues. Students in a graduate software engineering class in the College of

Information Science and Technology at Drexel University were directed to take a survey form to their place of employment and gather the required data. The data analyzed were from a small sample, 33 surveys, but were sufficient to show some interesting traits.

The survey was divided into four major categories of interest: 1) demographic; 2) market information; 3) organizational information; and 4) software quality. The main motivation for this survey was to examine information technology workers in terms of their views on quality, both in the aggregate and in subsets of managers versus technical staff.

The first stage of analysis was to view the demographic layout of the respondents. This was done very easily by simply looking at the sizes of the bins within the separate variables. Within the Demographics Section (for the sake of this paper) the major questions looked at were: 1) Type of company; 2) Number of employees in company; 3) Does the IS department work in teams; 4) What is the main purpose of the IS department within the organization; 5) The number of years of experience of the respondent; and 6) Whether they consider themselves technical or managerial. The initial MOTC graph is shown in Figure 25. The meaning of the bins is available interactively in MOTC. For example, Figure 26 is a display showing the meaning of the bins in the TypeCompany variable (or dimension).

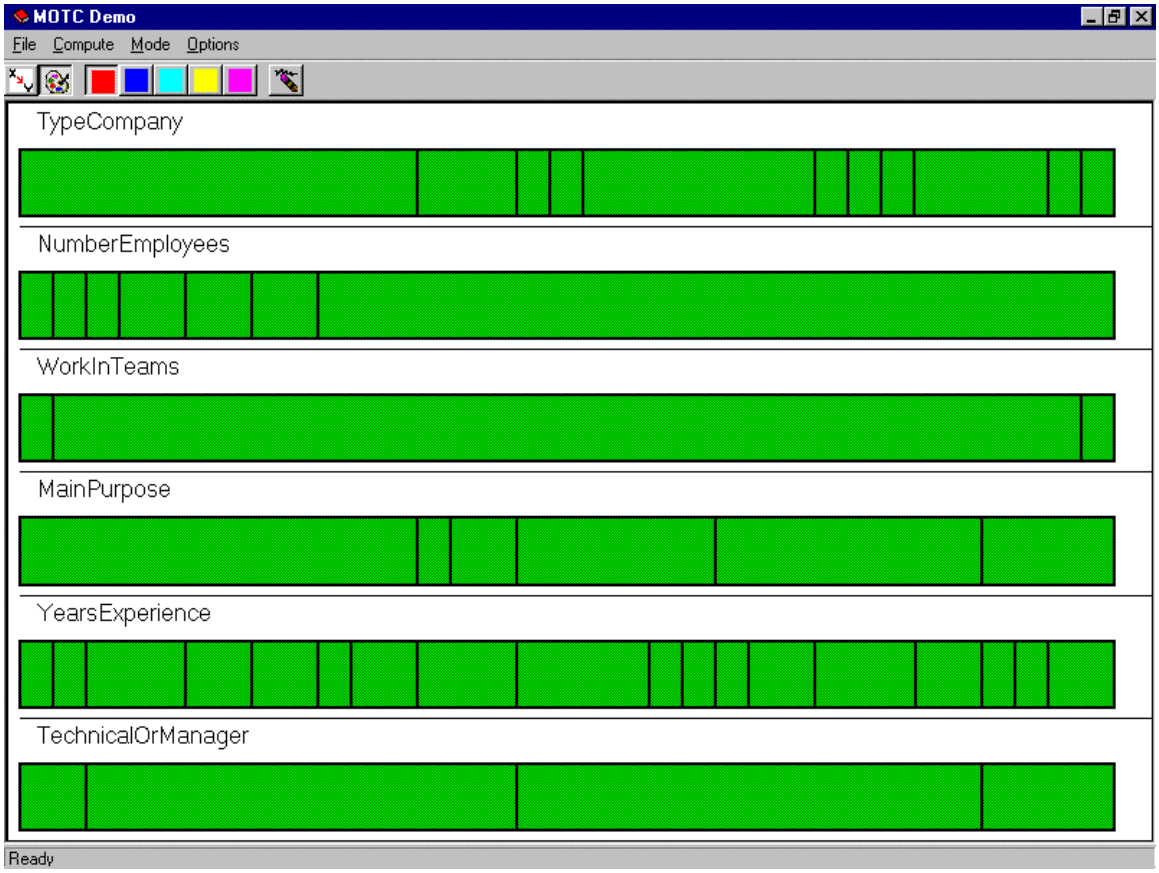


Figure 25. Initial MOTC View for Survey Data: Demographic Section

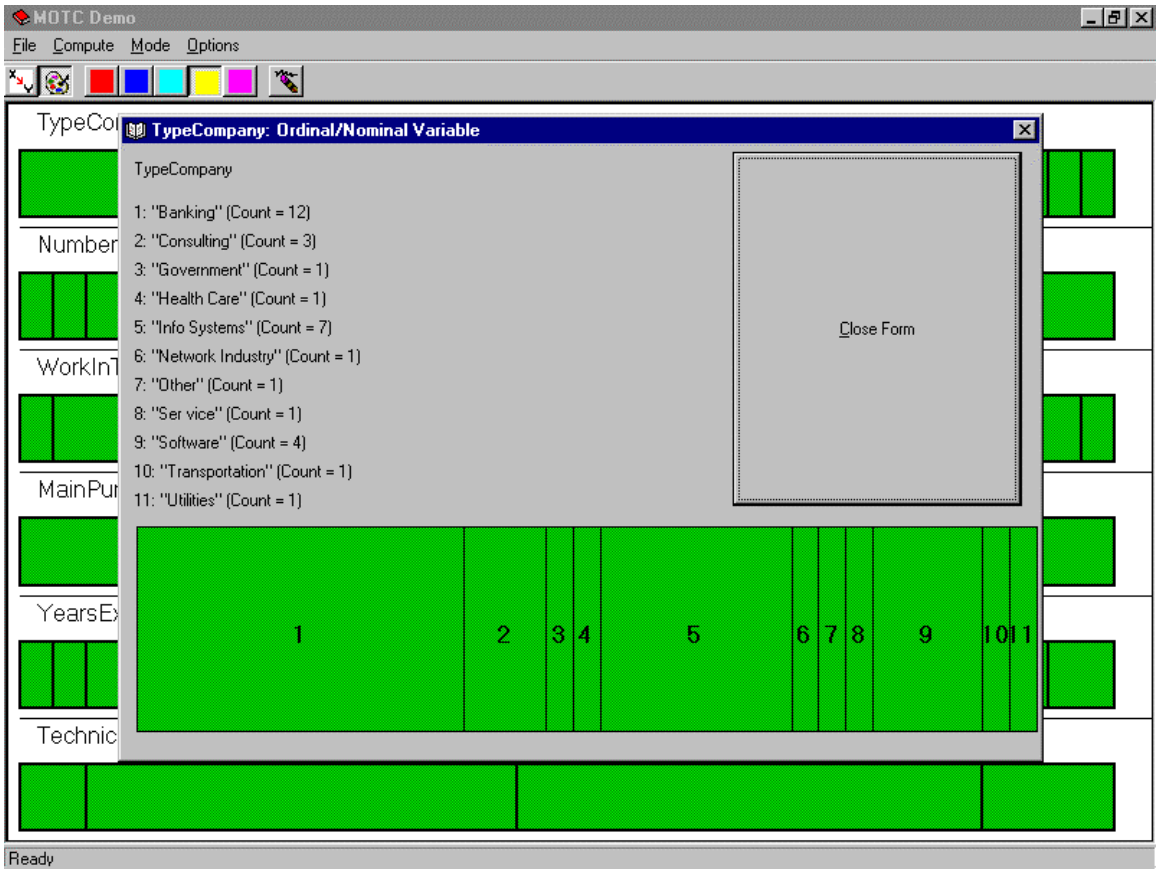


Figure 26. Listing of Categories by Bin for Dimension (variable) Type in Figure 25

It is easy-from Figures 25 and 26, and from MOTC in general-to see that three types of companies somewhat dominated in the survey. These were from "banking" (bin 1, 12 instances), "information systems" (bin 5, 7 instances) and "software" (bin 9, 4 instances) with the first category, "banking" being the largest. For the question on the number of employees, the last category, "Greater than 1000," was the biggest contributor. The majority of the respondents work in teams as shown by the categories, "Missing," "Yes," and "No." The main purposes of the departments described were mixed, but a large element was the first category "Maintenance." The years of experience of the respondent was very uniformly distributed (this is the only category that was not categorical). And finally, it is also easy to see that the numbers of "technical" versus "managerial"

workers were evenly mixed across the four categories: “Missing,” “Technical,” “Managerial,” and “Other.”

In using the brushing mode for demographics, we found some somewhat interesting possibilities; however, due to the small sample size, which sometimes yielded a bin with only one value, we regarded these patterns as spurious, and indeed overall they yielded no significant results. The major focus of this study was, after all, on quality issues, and we examined these next.

The section of the survey which dealt with software quality had various questions concerned with the issues of quality at large. We focused on one question for this paper. The particular question had the respondent pick four of twenty-two commonly used categories to describe quality, and then to rank the four chosen, with the one ranked the highest indicated in Quality Issue Rank 1, the next highest of the four ranked in the category Quality Issue Rank 2, and so on. The initial MOTC graph is as shown in Figure 27.

The initial MOTC view of just the distribution of responses yields some interesting trends. Of the first ranked quality, of the twenty-two possibilities, only nine were chosen, with “Functionally Correct” being the largest category. The later ranked categories were more splintered, having 12, 13, and 13 different choices, respectively. The category of Technical or Manager was included to allow the analyst to see differences between managers or technical people and how they view quality. This is very easily accomplished within MOTC simply by clicking the individual categories for managers and technical staff using a different color for each. Looking at the distribution of the quality issues by rank we get the display shown in Figure 28.

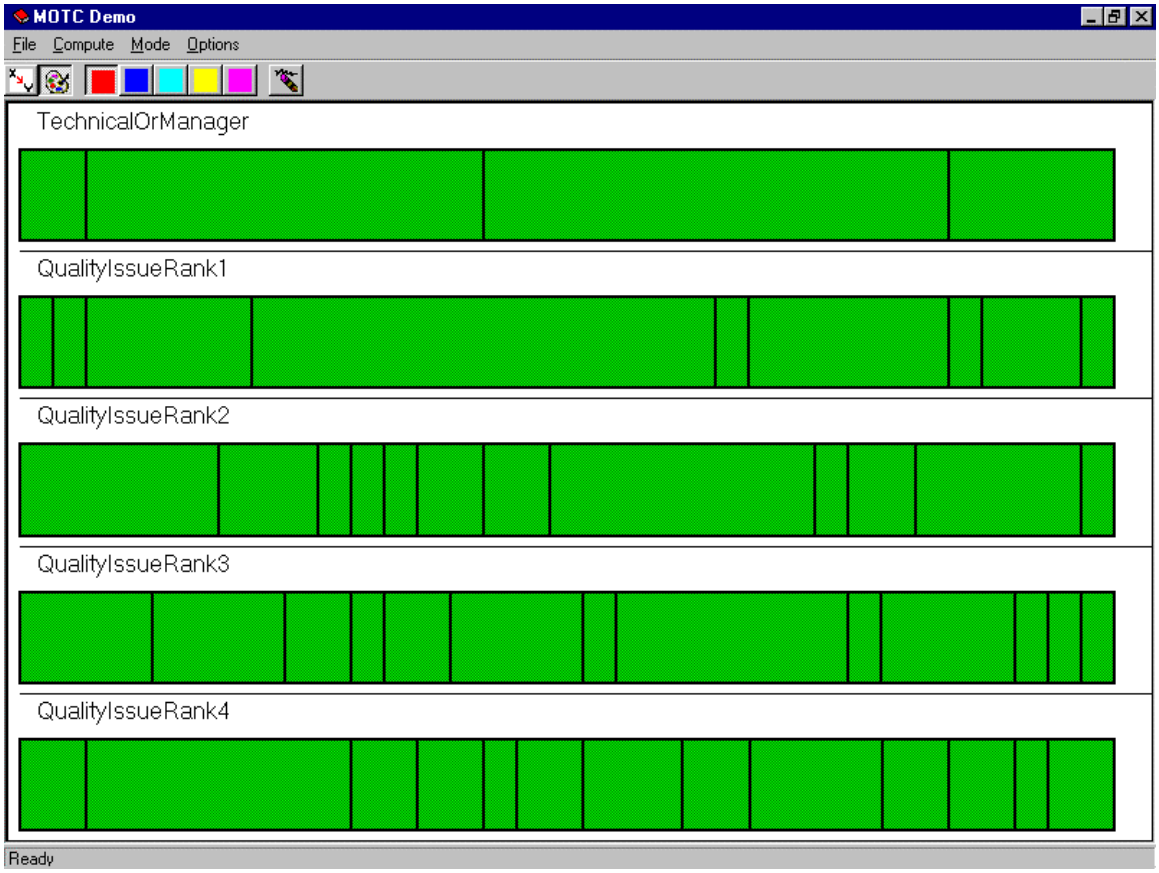


Figure 27. Initial MOTC View for Survey Data: Quality Section

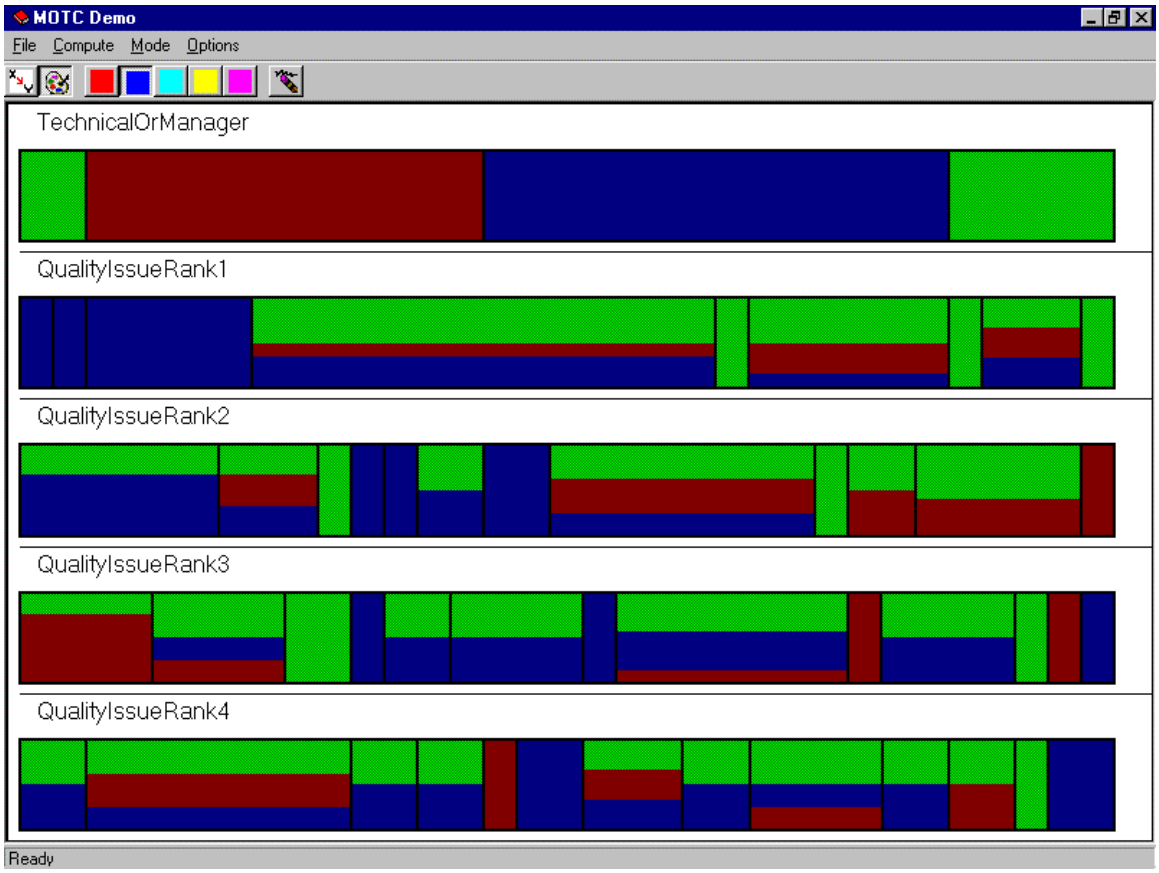


Figure 28. Brushed MOTC View for Survey Data: Quality Section

Once the technical staff and managers are brushed red and blue (so to speak), it is interesting (and very easy!) to see the somewhat different issues which appeal to each. If each group both chose the same category of quality, then the bin would reveal both colors, as in the eighth category of the first ranked quality issue, "usable." Of the choices by the technical people from the twenty-two possibilities, they felt only three should be ranked first, with those categories being "Functionally Correct," "Reliable," and "Usable." Several managers felt the same way. What is fascinating is that the first three categories, "Resilient," "Testable," and "Correct" were chosen only by the managers as most important, revealing a somewhat different philosophy.

We say that two categories are *clustered* if respondents who choose one category are then much more likely to also choose the other. To see if any categories are clustered, one can click on one of the first ranked responses, say "Functionally Correct," and then see if there are any other single categories that stand out in the lower rankings, say "Reliable." We find this from the MOTC diagram, as shown in Figure 29.

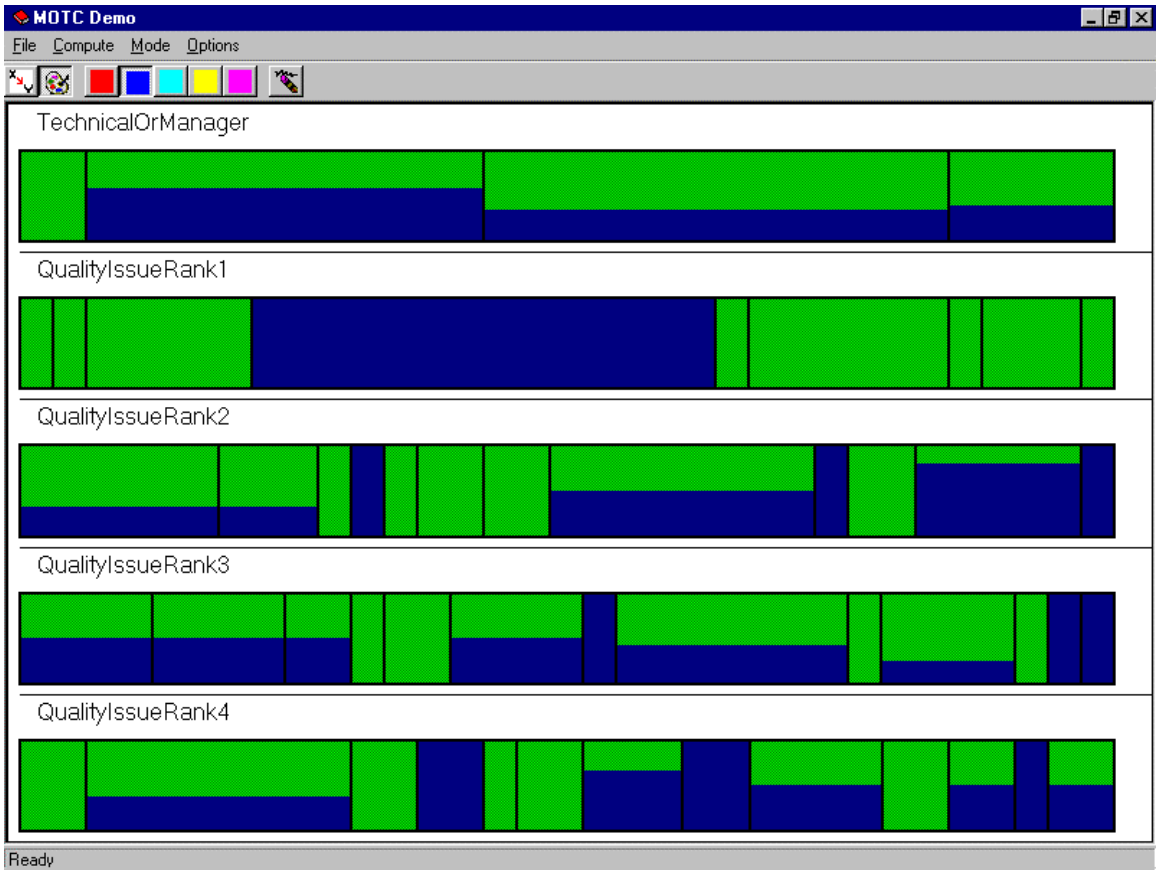


Figure 29. Brushed MOTC View for Survey Data: Clustered Categories

From this, one can see that choosing the fourth category, "Functionally Correct," yields no clustered category at the lower rankings, but instead is rather evenly distributed in lower rankings and even the within the job type. Clicking similarly on the other categories yielded the same results, so there is no real clustering, at least in the two dimensional space.

In conclusion, we found that using MOTC facilitated both the initial viewing of the distribution of the respondents, as well as the manipulations to determine if any simple association (or clustering) rules exist.

5.3 ANALYSIS OF OPTIMIZATION RESULTS

The Blue Ridge linear program is an example from an introductory operations research textbook [Ragsdale, 1995,pages 195-205]. The model is simple, with two decision variables and three structural constraints.

$$
\begin{array}{lll}
\max & 350 \cdot A + 300 \cdot H & \\
\text{s.t.} & & \\
 & 1 \cdot A + 1 \cdot H & \leq 200 \\
 & 9 \cdot A + 6 \cdot H & \leq 1520 \\
 & 12 \cdot A + 16 \cdot H & \leq 2650 \\
 & A, H & \geq 0 \\
 & A, H & \in \text{integer}
\end{array}
$$

where

A = the number of Aspas (a type of hot tub) to produce

H = the number of Hluxes (a type of hot tub) to produce

Although optimality for this model can very easily and quickly be determined by hand, we approached this example as a case for *candle-lighting*

analysis [Branley et al., 1997]. For which combined values of A and H does a high objective function value (also known as: absolute fitness) result? For purposes of this demonstration, the decision space was heuristically mapped by a genetic algorithm and the best 300 solutions were saved. These constituted our data. The MOTC analysis on these data included creating a MOTC binned bar set with three variables (A, H, and AF (absolute fitness, i.e., the value of the objective function)), where each bar was divided into eight bins.

In the data set we used, the best solutions had AF values ranging from 57,800 to 63,600. Only 8 of the 300 solutions had values in the top half (i.e., between 60,700 and 63,600), and only 2 were in the top bin (62,875 to 63,600). By clicking on bins 5 through 8 (in exploratory mode), we find that all of the solutions fall into bins 5, 6, 7, and 8 for variable A, and bins 2, 4, 5, and 6 for variable H. Moreover, *all* of the values in bin 7 for A are i n the top half of AF, and the majority of those in bin 5 corresponded to the top half of AF (Figure 30).

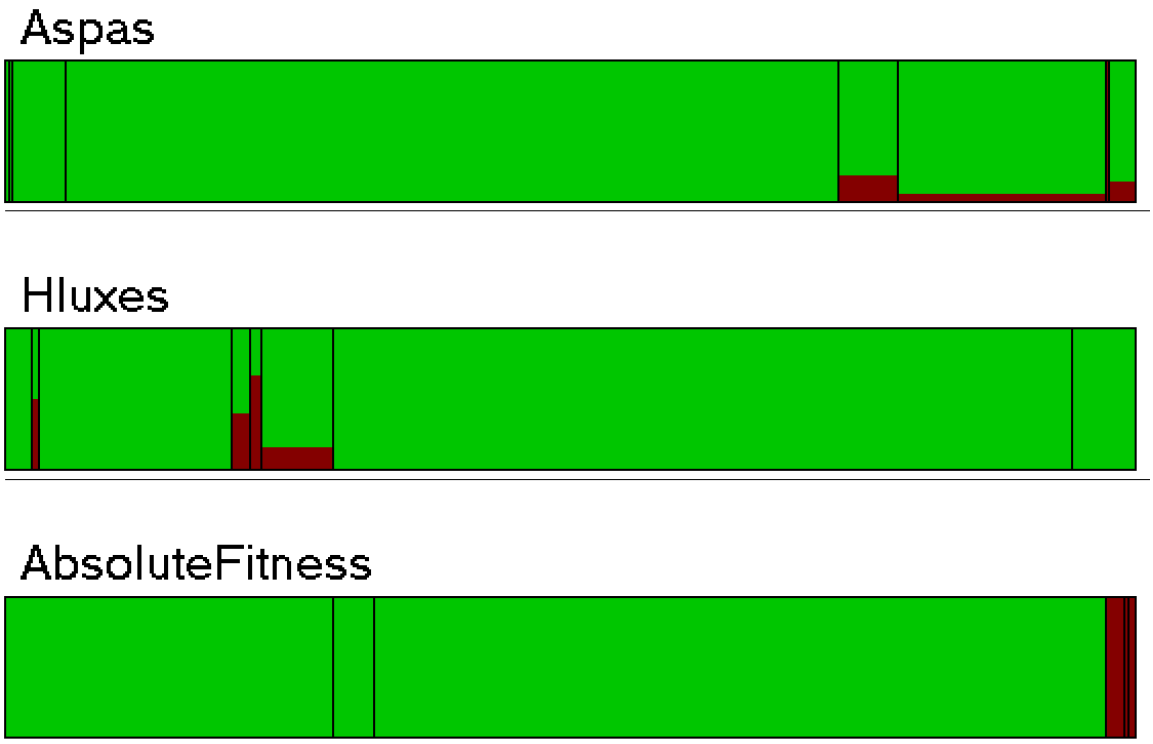


Figure 30. MOTC in Hypothesis Hunting Mode: Bins 5, 6, (7), and 8 of AF Bar Clicked

By further restricting our view to bin 8 of AF, we observe that both observations came from bin 5 of H, and one each from bins 6 and 7 of A (Figure 31).

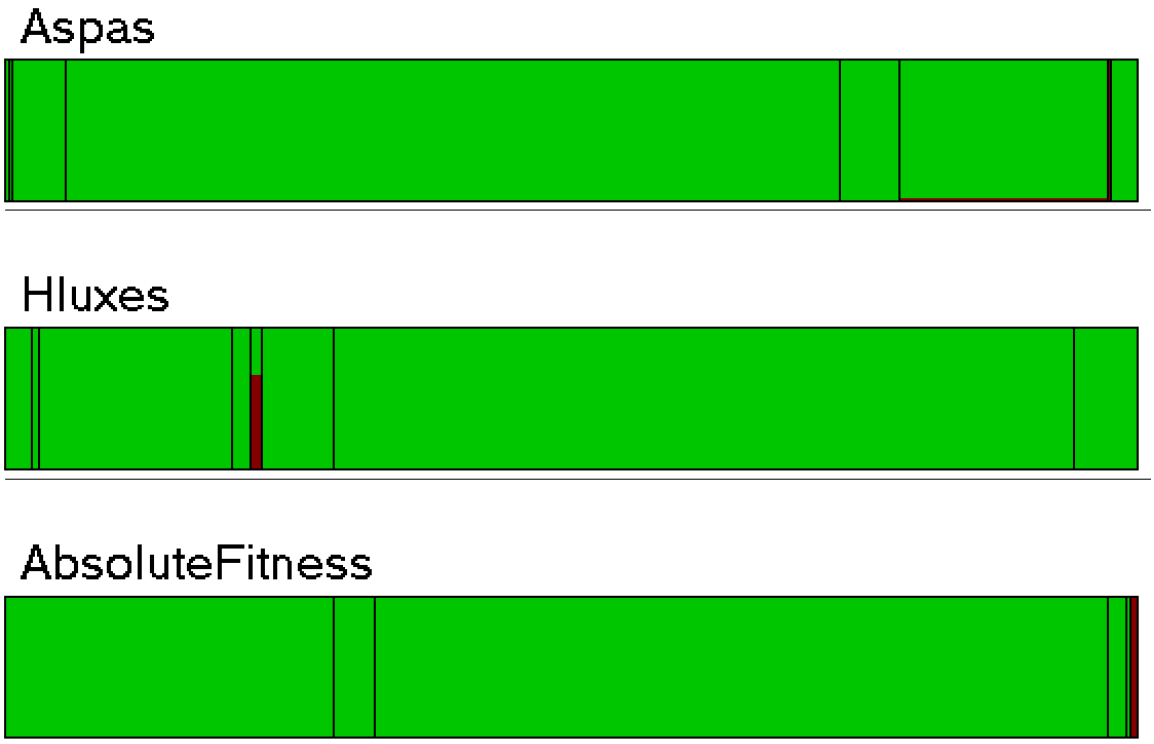


Figure 31. MOTC in hypothesis hunting mode: Bin 8 of AF bar clicked

Thus, after this exploration, a reasonable guess at a good hypothesis that we might make from this information is:

$P2$: H_5 ~>AF_8 and ($A_5 \vee A_6 \vee A_7 \vee A_8$) $\wedge$ ($H_2 \vee H_4 \vee H_5 \vee H_6$) ~> ($AF_5 \vee AF_6 \vee AF_7 \vee AF_8$)

This hypothesis, arrived at in just a few minutes using MOTC, has a ∇ value of 0.861, which is quite high.[9]

[9] As mentioned above, Prediction Analysis does have a significance testing theory, but that is outside the scope of this paper.

VI. COMPARISON WITH ALTERNATIVES

MOTC, as we have seen, assumes two main frameworks (the cross-tabulation form for representing hypotheses, and Prediction Analysis for measuring goodness of hypotheses), and provides an interactive environment of some promise for discovering interesting hypotheses. Here we want to consider the question, How does MOTC, or the ideas it embodies, compare with what has appeared in the relevant literature? The KDD literature has recently produced a number of interesting visualization approaches, e.g., [Ankerst et al., 1999,Goan and Spencer, 1999,Kichiyoshi et al., 1999,Ong and Lee, 1997], as well as the beginning of use of such approaches for rule discovery, e.g., [Gilbert et al., 1998,Imielinski et al., 1999,Lie et al., 1999]. Multidimensional databases are also gaining attention, e.g., [Goil and Choudhary, 1998], and other interactive tools for discovery are beginning to be reported, e.g., [Davidson et al., 1998]. Two points, however: (1) MOTC is unusual among database mining tools in using crosstabulation forms,[10] and (2) MOTC is unique in being an end-user interactive tool for supporting Prediction Analysis. For these reasons, we are less concerned in this section with establishing originality and are more focused on placing MOTC within the nexus of data visualization techniques. This serves the purposes of better understanding what MOTC is about and of pointing towards future research.

6.1 DESIGN GOALS OF THE MOTC INTERFACE

Stepping back and looking at the larger picture, the purpose of MOTC is to help the user discover interesting patterns in data and to provide an evaluation of the predictive value of those patterns. To this end, we identified three main desiderata for MOTC's interface design.

1. Present a display that can represent a very large number of records.

[10] Thanks to Balaji Padmanabhan for this point. See also [Moore and Lee, 1998,Ong and Lee, 1997].

The simple fact is that modern databases are huge and we need tools for dealing with them. Of course, for purposes of pattern discovery it is always possible-even desirable-to sample from the underlying data. Even so, having the option of examining larger datasets is always a good thing, since patterns evident in large datasets may not be apparent in smaller sample sets.

2. Effectively display a large number of variables.

   It is also a simple, or brute, fact that modern databases present large numbers of dimensions, or fields, among which users have an interest in discovering patterns. To limit a user's view of the data to only a subset of the data's variables is a severe restriction on the user's ability to discover patterns. Unfortunately, too many variables (dimensions) in a display can quickly overwhelm a user's cognitive resources. Therefore, a second goal of MOTC's interface is to maximize the number of displayed dimensions without overwhelming the user.

3. Provide for visualization that helps users discover associations among variables.

   Passively displaying information only goes so far in helping users discover patterns in the data. To be a truly effective interface the display must actively highlight associations among variables in the data by providing users with feedback about the quality of the apparent associations.

These are general goals, goals that have attracted study outside the context of MOTC. We now briefly review and discuss this literature.

6.2 PRESENT A DISPLAY THAT CAN REPRESENT A VERY LARGE NUMBER OF RECORDS

It is generally accepted that people more easily process visual information than textual or numerical information. “Scanning a thousand tiny bars with your eyes requires hardly any conscious effort, unlike reading a thousand numbers, which takes a great deal of mental energy and time” [Rao, 1997]. Information

visualization techniques can take advantage of this fact by displaying enormous amounts of information on the screen. For example, the SeeSoft system effectively displays over 15,000 lines of code on the screen [Eick et al., 1992] by representing code with pixel-thin lines that reflect the code's visual outline. InXight's "wide widgets" [Rao, 1997] are visual components that can be incorporated into a GUI information system to display several orders of magnitude more data than traditional display tools (e.g., spreadsheets or hierarchical trees). Wide widgets are focus+context interfaces [Furnas, 1986, Spence and Apperley, 1982] which dynamically distort spatial layouts so that users can zoom in on several records or variables while the rest of the records shrink to fit within the remaining space. In this way, users can focus on several items without losing the context provided by the remaining items. One wide widget, the Table lens, has been demonstrated with a table of baseball statistics containing 323 rows by 23 columns = 7429 cells [Rao and Card, 1995]. Others include the Perspective Wall [Mackinlay et al., 1991] and the Hyperbolic Tree Viewer [Lamping et al., 1995].[11]

Wright [Wright, 1997] demonstrates several applications that make use of 3D effects. One application, a financial portfolio manager displays more than 3,000 bonds on a single screen. This system uses color to indicate long and short positions, height for the bond's value, and the x and y axes to represent sub-portfolios and time to maturity.

Unfortunately, these techniques will fall short for very large databases, because, ultimately, we are limited to the number of pixels on the screen. Even with techniques like VisDB's pixel-oriented approach [Keim, 1996,Keim and Kriegel, 1994], which displays a data record per pixel, we are still limited to the number of pixels on the screen. With today's technology, this means approximately 1024 ×

[11] Images of these systems can be found on InXight's home page at http://www.inxight.com/vizcontrols.

1024 ≈ 1MB records which will not do for multi-million, gigabyte, and certainly not terrabyte-sized databases.

To present an unlimited number of records on the screen at once we need to present summaries of the data. If summaries are provided for each variable, then the only limitation is the number of variables that can be displayed regardless of the number of records in the database. The InfoCrystal [Spoerri, 1993] uses an innovative extension of Venn diagrams to visualize data summaries. MineSet's Evidence Visualizer [Becker, 1997] uses rows of pie charts to summarize the data. One row for each variable, one pie chart for each attribute. The pie chart represents the number of records matching the query variable's chosen value with the pie chart's value.

The approach of presenting summaries of *all* the data is strongly endorsed by Ben Shneiderman who preaches the following mantra (as he calls it) for designing visual information seeking systems:

> Overview first, zoom and filter, then details-on-demand. [Shneiderman, 1996,p. 2]

To overview very large numbers of records, we must sample or summarize. MOTC represents a summarization strategy (the cross-tabulation form), but there is nothing to prevent applying MOTC to sampled data.

6.3 EFFECTIVELY DISPLAY A LARGE NUMBER OF VARIABLES

The problem of displaying multidimensional data in an effective manner, one comprehensible to users, has been studied for some time (see [Jambu, 1991,Jones, 1995] for useful reviews). Perhaps the most natural and widespread approach for adding dimensions to a display is to add visual cues to an existing display. For example, the three dimensions of a 3D graph can be augmented by encoding points on the graph with color, texturing, shapes (glyphs), shading, and other such techniques. Becker [Becker, 1997] demonstrates the use of such techniques with the MineSet system, and various forms of these techniques are

supported by contemporary data visualization software tools (e.g., Advanced Visual Systems).

This family of techniques has two important limitations. First, there are only so many visual cues that can be employed. Perhaps 5-10 variables can be represented on a 2D display using the 3 geographic dimensions, color (divided into hue, saturation and brightness), shape, size, texture, and shading. Second, and more limiting, is that humans cannot effectively process that many visual cues of this sort at once. More than a few visual cues quickly overwhelm users. Projecting multiple dimensions onto a two-dimensional plane also becomes quickly illegible. Jones [Jones, 1995,Chapter 14], for example, reports that 8 dimensions is too much for this technique and even 6 and 7 dimensions are difficult to comprehend.

As an example, Feiner and Beshers' Worlds Within Worlds technique [Feiner and Beshers, 1990], which plots n dimensions by successively embedding 3-dimensional coordinate systems inside one another, can theoretically display any number of dimensions on the screen. However, Jones [Jones, 1995,Chapter 14] points out that more than three levels (9 dimensions) is incomprehensible and even 2 levels (6 dimensions) can be difficult to assimilate.

In MOTC, we present the same visual cue for each variable (a horizontal bar on the screen, with coloring), and use secondary visual cues (position, color) to distinguish the categories associated with a variable (the bins). A popular set of techniques using this approach are graphical matrices in which rows and columns represent variables, and each cell in the matrix is a comparison of the pair of variables represented by the cell's row and column. Perhaps the most common representation of the two variables associated with a matrix cell is a scatter plot [Jones, 1995, Becker et al., 1987, Becker and Cleveland, 1987]. However, other representations are possible, such as histogram profiles [Tweedie et al., 1996], boxplots and sunplots [Jambu, 1991,Chapter 5].

Unfortunately, graphical matrices only allow direct comparisons between two variables. A simpler technique is to display a row of variables. When combined with brushing (see above), variable rows allow any number of variables to be directly compared. MineSet's Evidence Visualizer [Becker, 1997], with its rows of pie charts, does just this. The Influence Explorer [Tweedie et al., 1996] presents rows of histograms, each histogram summarizing the values of a single variable. Thus, MOTC's display approach for variables should, in future research, be assessed as a member of this category of representation. Very likely it will be possible to improve the display, but that is something to be determined by extended empirical testing, something that has yet to be done for nearly all the interesting techniques.

Even using graphical matrices of variable rows, the number of variables that can be displayed is limited to the number of rows or columns that can fit on the screen. A natural extension of this technique to use the focus+context ability of Table Lens [Rao and Card, 1995] to augment the number of rows and columns displayed, thereby augmenting the number of variables. Indeed, the interface for MOTC is a crude example of this idea: the underlying dataset can have a very large number of dimensions, among which the user picks up to either for a particular analysis; different dimensions can be picked in different analyses. In future editions of MOTC (or MOTC-like systems), we would think that this process could be made smoother and easier and that doing so would benefit the user.

One more technique is worth noting. Inselberg's parallel coordinates system [Inselberg, 1985, Inselberg and Dimsdale, 1989, Jones, 1995] represents variables as vertical bars, and database records as “polylines” which connect each of the variables' vertical bars. Where a polyline crosses a variable's vertical bar represents that polyline's record's value for the variable. This technique allows for a very large number of variables to be displayed-as many variables as vertical lines that will fit on the screen. The drawback of this approach is that

each polyline represents one record, so the technique is limited to displaying only a relatively small number of records.

6.4 VISUALIZING ASSOCIATIONS BETWEEN VARIABLES

Visualization techniques are known to be very helpful for discovering patterns in data. This is especially so for relationships between two variables. Things are more difficult when multiple variables are involved. For this problem, MOTC's approach is of a kind that is accepted in the literature: present multiple variables and support active display of linkages among them. For example, selecting a record or range of records in one of the Influence Explorer's histograms highlights the corresponding records in the other histograms [Tweedie et al., 1996]. Similarly, the Lifelines system [Plaisant et al., 1996] displays compact medical patient histories in which users can, say, click on a particular patient visit and immediately see related information, such as other visits by the same patient, medication, reports, prescriptions and lab tests. Visage [Kolojejchick et al., 1997, Roth et al., 1996] presents multiple views of the same data. One window may present geographic data in map form, while another window presents the data as a histogram, and yet another presents the data in a table. Selection of a subset of data in any window, highlights the corresponding representation of the data in the other windows. Graphical matrices can by dynamically linked through *brushing* [Becker and Cleveland, 1987,Becker et al., 1987] in which selecting a set of records in one scatterplot (or whatever graphical technique is used for the graphical matrix) simultaneously highlights the same records in the rest of the matrix's cells.

MOTC's use of brushing (see above) should be seen as a visualization approach of the kind explored in this literature. As with the issue of display of multiple dimensions, much further research is needed in order to find the optimal design (if there is one) of this sort.

VII. SUMMARY AND DISCUSSION

So, what have we got and how good is it? Recall that earlier we argued for a series of goals for any tool to support the hypothesis generation activity in KDD and database mining. Here, with additional comments, is that list again.

1. *Support users in hypothesizing relationships and patterns among the variables in the data at hand.* MOTC has hypothesis hunting mode, in which users may use the mouse quickly and interactively to try out and test arbitrary hypotheses, and thereby explore hypothesis space.

2. *Provide users with some indication of the validity, accuracy, and specificity of various hypotheses.* MOTC employs Prediction Analysis for this.

3. *Provide effective visualizations for hypotheses, so that the powers of human visual processing can be exploited for exploring hypothesis space.* MOTC contributes an innovation in visualization by representing multidimensional hypotheses as binned bars that can be brushed with a mouse. Also, MOTC innovates by tying together hypothesis hunting and evaluation, and does so with a common visual representation.

4. *Support automated exploration of hypothesis space, with feedback and indicators for interactive (human-driven) exploration.* MOTC does not do this at present, although we have plans to add these features. Briefly, we intend to begin by using a genetic algorithm to encode and search for hypotheses (see Table 2). As in our candle-lighting work [Branley et al., 1997], we envision storing the most interesting solutions found by the genetic algorithm during its search and using these solutions as feedback to the user.

5. *Support all of the above for data sets and hypotheses of reasonably high dimensionality, say between 4 and 200 dimensions, as well as on large data sets (e.g., with millions of records).* MOTC is not computationally very

sensitive to the number of underlying records. We have worked successfully with much larger data sets than those we report here. But, MOTC is sensitive to the number of cells in the crosstab grid. With 10 variables and 10 bins per variable, the multi-dimensional data grid has 10^{10} cells, a number perhaps too large for practical purposes. On the other hand, 12 variables with only 4 bins each is only $4^{12} \approx 16$ million cells, and this is quite manageable on today's PCs. In short, MOTC-like systems will work over a wide range of useful and computationally feasible problems.

All of this, we think, looks very good and very promising. Still, the ultimate value of any system like MOTC has to be determined by testing real people on real problems. Our experience to date, which is admittedly anecdotal, is very encouraging. Moreover, we note that if you value Prediction Analysis, then you need to calculate ∇, U and so on. MOTC makes these calculations and does this quickly and easily from a user's point of view. All this is excellent reason to proceed to experiments with real people and real problems. But that is subject for another paper.

ACKNOWLEDGEMENTS

Note: This paper is an expanded version of “MOTC: An Aid to Multidimensional Hypothesis Generation” by K. Balachandran, J. Buzydlowski, G. Dworman, S.O. Kimbrough, E. Rosengarten, T. Shafer, and W. Vachula, in[Balachandran et al., 1999].

Special thanks to James D. Laing for introducing us to Prediction Analysis, for encouraging noises as these ideas were developed, and for insightful comments on an earlier version of this paper. Thanks also to Balaji Padmanabhan for some useful comments and suggestions. None of the shortcomings of this paper should be attributed to either Laing or Padmanabhan. Send correspondence to Steven O. Kimbrough at the address in the heading. File: motccaisBill.doc from motcjmis.tex, from MotcHICS.TEX. This material is

based upon work supported by, or in part by, the U.S. Army Research Office under contract/grant number DAAH04-1-0391, and DARPA contract DASW01 97 K 0007.

EDITOR'S NOTE: This article was received on November 6, 2000 and was published on December 31,2000.

REFERENCES

EDITOR'S NOTE: The following reference list contains hyperlinks to World Wide Web pages. Readers who have the ability to access the Web directly from their word processor or are reading the paper on the Web, can gain direct access to these linked references. Readers are warned, however, that
1. these links existed as of the date of publication but are not guaranteed to be working thereafter.
2. the contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
3. the author(s) of the Web pages, not AIS, is (are) responsible for the accuracy of their content.
4. the authors of this article, not AIS, are responsible for the accuracy of the URL and version information.

Ankerst, M. M. et al. (1999) "OPTICS: Ordering Points to Identify the Clustering Structure", *SIGMOD Record*, (28)2, pp. 49-60

Balachandran et al. (1999) "MOTC: An Interactive Aid for Multidimensional Hypothesis Generation", *Journal of Management Information Systems*, (16)1, pp.17-36

Becker, B. G. (1997) "Using Mineset for Knowledge Discovery", *IEEE Computer Graphics and Applications*, July/August, pp. 75-78

Becker, R. A. and W. S. Cleveland (1987) "Brushing Scatterplots", *Technometrics*, (29)2, pp. 127-142

Becker, R. A. et al. (1987) "Dynamic Graphics for Data Analysis", *Statistical Science*, (2)4, pp. 355-395

Branley, B. et al. (1997) “On Heuristic Mapping of Decision Surfaces for Post-evaluation Analysis” in Nunamaker, Jr., J. F. and Sprague, Jr., R. H. (eds.) (1997) *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, Los Alamitos, CA: IEEE Computer Society Press

Codd, E. F., S. B. Codd and C. T. Salley (1993) “Beyond Decision Support”, *Computerworld*, (27)30

Davidson, G. S. et al. (1998) “Knowledge Mining with VxInsight: Discovery through Interaction” *Journal of Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies*, (11)3, pp. 259-85

Dhar, V. and R. Stein (1997) *Seven Methods for Transforming Corporate Data into Business Intelligence*, Upper Saddle River, NJ: Prentice-Hall, Inc.

Eick, S.B., J.L. Steffen, and E.E. Sumner (1992) “Seesoft - a Tool for Visualizing Line Oriented Software”, *IEEE Transactions on Software Engineering*, (18)11, pp. 957-968

Fayyad. U.M. et al. (ed.) (1996) *Advances in Knowledge Discovery and Data Mining*, Cambridge, MA: The MIT Press

Feiner, S. and C. Beshers (1990) “Worlds within Worlds: Metaphors for Exploring N-Dimenstional Virtual Worlds” in *Proceedings of the ACM Symposium on User Interface Software 1990*, ACM Press, pp. 76-83

Furnas, G.W. (1986) “Generalized Fisheye Views”, In *ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, pp.16-23.

Gilbert, K., T. Aluja, and U. Cortes (1998) “Knowledge Discovery with Clustering Based on Rules: Interpreting Results” in *Principles of Data Mining and Knowledge Discovery. Second European Symposium, PKDD`98, Proceedings.*, Berlin, Germany: Springer-Verlag, pp. 83-92

Goan, T. and L.Spencer (1999) “Supporting Thorough Knowledge Discovery with Data-Aware Visualizations” in *Proceedings of the Second International Conference on Information Fusion: FUSION `99*, Mountain View, CA: International Society for Information Fusion, pp. 567-71

Goil, S. and A. Choudhary (1998) “High Performance Multidimensional Analysis and Data Mining” in *Proceedings of ACM/IEEE SC98: 10th Anniversary. High Performance Networking and Computing Conference*, Los Alamitos, CA: IEEE Computer Society, pp. 801 – 802

Hildebrand, D.K., J.D. Laing, and H.Rosenthal (1997a) *Analysis of Ordinal Data*, Newbury Pack, CA: Sage Publications

Hildebrand, D.K., J.D. Laing, and H.Rosenthal (1977b) *Prediction Analysis of Cross Classification*, New York, NY: John Wiley & Sons, Inc.

Imielinski, T., A. Virmani, and A. Abdulghani (1999) “Dmajor – Application Programming Interface for Database Mining”, *Data Mining and Knowledge Discovery*, (3)4, pp. 347-72

Inmon, W. H. (1996) *Building the Data Warehouse, 2nd Edition*, New York, NY: John Wiley & Sons, Inc.

Inselberg, A. (1985) “The Plane with Parallel Coordinates”, *The Visual Computer*, pp. 69-91

Inselberg, A. and B. Dimsdale (1989) “Visualizing Multi-variate Relations with Parallel Coordinates” in *Third International Conference on Human-Computer Interaction, Work with Computers: Organizational, Management, Stress and Health Aspects*, pp. 460-467

Jambu, M. (1991) “Exploratory and Multivariate Data Analysis” in *Statistical Modeling and Decision Science*, San Diego, CA: Academic Press, Inc.

Jones, C. V. (1995) “*Visualization and Optimization*”, Boston, MA: Kluwer Academic Publishers

Keim, D. A. (1996) “Pixel-oriented Visualization Techniques for Exploring Very Large Databases”, *Journal of Computational and Graphical Statistics*, March

Keim, D. A. and H Kriegel (1994) “VisDB : Database Exploration Using Multi-dimensional Visualization”, *IEEE Computer Graphics and Applications*, September, pp. 40-49

Kichiyoshi, K. et al. (1999) “Data Visualization for Supporting Query-based Data Mining” in *IEEE SMC `99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics*, (5), pp. 888-93

Kimbrough, S. O., J. Oliver and C. Pritchett (1993) “On Post-evaluation Analysis: Candle Lighting and Surrogate Models”, *Interfaces*, (23)3, pp. 17-28

Kolojejchick, J., S. F. Roth and P. Lucas (1997) “Information Appliances and Tools in Visage” in *IEEE Computer Graphics and Applications*, pp. 32-41

Lamping, J., R. Rao and P. Pirolli (1995) “A Focus+context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies” in Katz, I. R., R. Mack and L. Marks (eds.) *ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM Press

Lie, B. et al. (1999) “Visually Aided Exploration of Interesting Association Rules” in *Methodologies for Knowledge Discovery and Data Mining, Third Pacific-Asia Conference, PAKDD-99, Proceedings*, Berlin, Germany: Springer-Verlag, pp. 380-9

Mackinlay, J. D., G. G. Robertson and S. K. Card (1991) “The Perspective Wall: Detail and Context Smoothly Integrated” in *ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, pp. 173-179

Menninger, D. (1995) “Oracle OLAP Products: Adding Value to the Data Warehouse”, Oracle White Paper: Part #: C10281

Moore, W. and M.S. Lee (1998) “Cached Sufficient Statistics for Efficient Machine Learning with Large Datasets”, *Journal of Artificial Intelligence Research*, (8)3, pp.67-91

Hwee, L.O. and H.Y. Lee (1997) “A New Visualisation Technique for Knowledge Discovery in OLAP” in *Proceedings of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining. KDD: Techniques and Applications*, Singapore: World Scientific, pp. 279-286

Plaisant, C. et al. (1996) “Lifelines: Visualizing Personal Histories” in Tauber, M.J. (ed.) *ACM SIGCHI Conference on Human Factors in Computing Systems*. ACM Press,pp. 221-227

Ragsdale, C.T. (1995) *Spreadsheet Modeling and Decision Analysis: A Practical Introduction to Management Science*, Cambridge, MA: Course Technology

Rao, R. and S.K. Card (1995) “Exploring Large Tables with the Table Lens” in Katz I. R., R. Mack, and L. Marks (eds.), *ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM Press pp. 403-404

Rao, R. (1997) “From Research to Real World with Z-GUI” in Gershon, N. and S.G. Eick (eds.) *IEEE Computer Graphics and Applications*, pp.71-73

Rao, R. and S.K. Card (1994) “The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+context Visualization for Tabular Information” in *Proceedings of the CHI `94 Conference*, pp. 318-323

Roth, S.F et al. (1996) “Visage: A User Interface Environment for Exploring Information” in *IEEE Conference on Information Visualization*, IEEE Computer Press, pp. 3-12.

Shneiderman, B. (1996) “The Eyes Have it: A Task by Data Type Taxonomy of Information Visualizations” in *Proceedings of the IEEE Symposium on Visual Languages 1996*, IEEE Publications, pp. 336-343. (An active hyperlinked version of Shneiderman's taxonomy is available at the OLIVE site http://otal.umd.edu/Olive.)

Spense, R. and M. Apperley (1982) “Data Base Navigation: An Office Environment for the Professional”, *Behaviour & Information Technology*, (1)1, pp. 43-54

Spoerri, A. (1993) “Infocrystal: A Visual Tool for Information Retrieval & Management” in *Information Knowledge and Management '93*, November

Tauber, M. J. (ed.) (1996) *ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM Press

Tweedie, L. A. et al. (1996) "Externalising Abstract Mathematical Models" in Tauber, M. J. (ed.) *ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM Press

Wand, M. P. (1997) "Data-based Choice of Histogram Bin Width", *The American Statistician*, (51)1, pp. 59-64

Wright, W. (1997) "Business Visualization Applications", *IEEE Computer Graphics and Applications*, pp. 66-70

ABOUT THE AUTHORS

Krishnamohan Balachandran received his Ph.D. at the Operations and Information Management Department of the Wharton School of the University of Pennsylvania. He received a B. Tech from the Indian Institute of Technology, Bombay, in 1993 and an M.S. from the Systems Engineering Department of the University of Pennsylvania in 1995. His current research interests are benchmarking, consumer behavior, and data-mining applications related to these fields.

Jan Buzydlowski holds a bachelor's degree in mathematics and master's degrees in statistics and computer science. His interests are in data storage and retrieval and in knowledge discovery. He is currently finishing his Ph.D. in information Systems at the College of Information and Technology at Drexel University.

Garrett Dworman received his Ph.D. from the Department of Operations and Information Management Department of the Wharton School of the University of Pennsylvania. The main theme of his research is the design of cognitively

motivated information access systems. In his dissertation he developed pattern-oriented systems for accessing document collections in collections in museums and the health-care industry.

Steven O. Kimbrough is a Professor at the Wharton School of the University of Pennsylvania. He received his Ph.D. in philosophy from the University of Wisconsin-Madison. His main research interests are electronic commerce, information management, logic modeling, and rationality of computational agents. His active research areas include computational approaches to belief revision and non-monotonic reasoning, formal languages for business communication, agents, games and evolution, and information management.

Tate Shafer has a B.S. from the Wharton School of the University of Pennsylvania, where he studied information systems and finance. He served as an undergraduate research assistant to Steven O. Kimbrough and researched various IS fields, including computer programming, artificial intelligence, and information retrieval. He served for three semesters as a teaching assistant for the introductory information systems course at the Wharton School. He now works for an investment banking firm in New York.

William Vachula is currently a Ph.D. candidate in the Operations and Information Management department at The Wharton School of the University of Pennsylvania, focusing within its Information and Decision Technology discipline. Mr. Vachula received a BSEE from Carnegie-Mellon University (1983) and an MSEE from the University of Pennsylvania (1989). His industry experience includes software and systems engineering, project and program management, and information systems consulting. His research interests embrace various application domains for software agents, business communication languages, computational economics techniques, and advanced systems analysis and design processes.

Communications of the Association for Information Systems

ISSN: 1529-3181